S-100 - Part 16 Interoperability Catalogue Model

Contents

| Contents | 2 |
|--|----|
| 16-1 Scope | 6 |
| 16-2 Conformance | 6 |
| 16-3 Normative References | 6 |
| 16-4 Context | 6 |
| 16-4.1 Overview of approach to implementing interoperability | 6 |
| 16-4.2 Overview of an interoperability catalogue | 7 |
| 16-4.3 Overview of processing | 9 |
| 16-4.4 Interoperability Catalogue Data Model | 10 |
| 16-4.4.1 Introduction | 10 |
| 16-4.4.2 Interoperation Conceptual Types | 13 |
| 16-4.4.2.1 S100_IC_InteroperabilityCatalogue | 13 |
| 16-4.4.2.2 S100_IC_DisplayPlane | 14 |
| 16-4.4.2.3 S100_IC_Feature | 16 |
| 16-4.4.2.4 S100_IC_DrawingInstruction | 17 |
| 16-4.4.2.5 S100_IC_SuppressedFeatureLayer | 18 |
| 16-4.4.2.6 S100_IC_PredefinedCombination | 19 |
| 16-4.2.2.7 S100_IC_FeatureDerivation | 20 |
| 16-4.2.2.8 S100_IC_SuppressedFeatureInstance | 21 |
| 16-4.2.2.9 S100_IC_HybridFeature | 22 |
| 16-4.2.2.10 S100_IC_HybridFeatureCreationRule | 22 |
| 16-4.2.2.11 S100_IC_CompleteRule | 23 |
| 16-4.2.2.12 S100_IC_ThematicRule | 23 |
| 16-4.2.2.13 S100_IC_SimpleRule | 24 |
| 16-4.2.2.14 S100_IC_HybridFC | 24 |
| 16-4.2.2.15 S100_IC_HybridPC | 24 |
| 16-4.2.2.16 FeatureSelector | 25 |
| 16-4.2.2.17 requirementType | 25 |
| 16-4.4.2.18 dataProduct | 26 |
| 16-4.2.3 Filters | 28 |
| 16-5 Correspondence to and supersession of portrayal catalogue elements | 28 |
| 16-6 Feature creation or replacement rule | 29 |
| 16-7 Interoperability Catalogue Data Model and the General Feature Model | 31 |
| 16-7.1 Geographic feature types | 31 |
| 16-7.2 Meta feature types | 32 |
| 16-7.3 Feature and Information associations | 32 |
| 16-7.4 Information Types | 32 |
| 16-8 Interoperability Levels | 32 |

| 16-9 Hybridization rules | 32 |
|---|----|
| 16-9.1 Simple hybridization rule | |
| 16-9.2 Thematic hybridization rule | |
| 16-9.3 Complete hybridization rule | |
| 16-9.3 Hybrid Feature and Portrayal Catalogues | |
| 16-10 Production Process of Interoperability Catalogues | |
| 16-10.1 Management of Feature Catalogue and Portrayal Catalogue updates | 34 |
| 16-10.2 Product Specification updates other than FC/PC | 35 |
| 16-10.3 Unpredictable Product Specification updates | 35 |
| 16-10.4 Additional data products to be defined in catalogue | 35 |
| 16-10.5 Backward compatibility | 35 |
| 16-10.6 Interoperability Catalogue product | |
| 16-10.6.1 Updating the Interoperability Catalogue | |
| 16-10.6.2 Cancelling a version of the Interoperability Catalogue | |
| 16-10.7 Support files | 37 |
| 16-10.7.1 Updating the Interoperability Catalogue support files | 37 |
| 16-10.7.1.1 New Edition of the support files | 37 |
| 16-10.7.1.2 Revisions to the support files | 37 |
| 16-10.7.1.3 Clarification to the support files | 37 |
| 16-11 Portrayal | 37 |
| 16-11.1 Display of significant features | |
| 16-11.1.1 Switching to original | |
| 16-11.2 Impacts on viewing groups | |
| 16-11.3 Impacts on Portrayal Catalogues | |
| 16-11.4 Meta-features | |
| 16-11.4.1 Data quality for individual products | |
| 16-11.4.2 Portrayal of data quality for combinations | |
| 16-11.5 Display of text | |
| 16-11.6 Blended feature concepts or blended portrayals | |
| 16-11.7 Blended portrayal's effect on symbols and area patterns | |
| 16-11.8 Hierarchy of data | |
| 16-11.9 Interacting gridded information | 40 |
| 16-11.10 Pick reports | 40 |
| 16-11.11 User control over loaded set | 40 |
| 16-11.12 User control over interoperation level | 40 |
| Appendix 16-A | 41 |
| Appendix 16-B | 51 |
| Appendix 16-C | 58 |
| Appendix 16-D | 63 |

| Appendix 16- |)-E | 67 |
|--------------|-----|----|
|--------------|-----|----|

16-1 Scope

Users of S-100-based data products are likely to receive and use different S-100-based data products simultaneously and those data products are likely to contain features which may have some common elements or be different expressions of similar underlying concepts. Other data layers such as radar overlays may also be present. The smooth interoperation and harmonized, user-friendly graphical presentations of these various products is necessary. This Part defines a framework for creating rules for the interoperation of S-100 data products, including harmonized graphical presentations and handling of alarms and indications.

This framework can be used to establish system specific rules which are contained in an interoperability catalogue, a type of meta-product that describes how groups of products are to be used and displayed simultaneously.

16-2 Conformance

This Part is not a profile of an ISO or other standard because the concept of S-100 interoperability as used in this Part is unique to S-100. The S-100 Interoperability Model is a specialization of the CT_Catalogue class defined in ISO 19139, as implemented in ISO 19115-3.

16-3 Normative References

RFC 2141 URN Syntax. Internet Engineering Task Force (IETF), May 1997.

- IMO MSC.1/Circ.1512 Guideline on Software Quality Assurance and Human-Centred Design for e-Navigation.
- ISO 19115-1 ISO 19115-1:2014, Geographic information Metadata Part 1 Fundamentals. As amended by Amendment 1, 2018.
- ISO 19115-3 ISO/TS 19115-3:2016, Geographic information Metadata XML schema implementation for fundamental concepts.
- XPath XML Path Language (XPath) 3.1 W3C Recommendation 21 March 2017 (World-Wide Web Consortium W3C). URL: <u>https://www.w3.org/TR/2017/REC-xpath-31-20170321/</u> (Retrieved 2017-12-08).

16-4 Context

This Part provides a framework for specifying machine-readable rules governing the interaction of data products in an S-100 compatible systems. A system that utilises different data products needs a prescribed algorithm to determine how these different data products should interact within the system. Interaction in this context means the manner in which the simultaneous display of data products affects the appearance of the combined display compared to the appearance of each product displayed in isolation. Interoperability of products includes issues such as ensuring that additional information overlay products do not obscure other significant information, or managing the presence of same or similar feature instances in different products, which can leave the user questioning which instance is the most accurate and appropriate for use.

The rules governing interoperability are not expected to be the same for all product interactions and therefore must be managed for groupings of products that are expected to be used together under specific circumstances, for example when performing specific navigational tasks. Rules that manage the information according to these principles, are captured in a machine-readable XML format, so that systems can be updated with new sets of rules in events such as additional products becoming available or alterations to the existing combinations becoming necessary. The set of rules is called an interoperability catalogue.

16-4.1 Overview of approach to implementing interoperability

Interoperability processing works in combination with regular portrayal processing (see S-100 Part 9 - Portrayal). Depending on the system architecture designed by developers of system software, it may for example function as a pre-processing or post-processing stage to regular portrayal processing.

An Interoperability Catalogue basically describes a transformation from an input stream of feature data to an output stream of prioritized feature data. The input stream consists of feature data from S-100 based datasets (either in the form of feature objects from the input data, or feature objects transformed to drawing instructions by portrayal processing). The output feature stream consists of feature data that may contain some input features in their original form; combine versions of other features; and remove other features from the stream altogether. Features in the output stream also have assigned (or revised) priorities in relation to other features, depending on feature type.

16-4.2 Overview of an interoperability catalogue

An Interoperability Catalogue is a collection of rulesets for filtering and/or combining feature objects from different input streams corresponding to different data products into feature objects in a combined output stream. The Interoperability Catalogue provides means of describing the conditions under which a ruleset is active – that is, each ruleset applies to a particular combination of products.

Within each ruleset, there may be rules whose antecedents specify the combinations of features from different products to which they apply, and what the result of applying the rule is; that is, given a particular combination of features in the input stream, what feature or features should be emitted into the output stream.

The interoperability processor applies the rules which are assigned to the loaded combination of data products, and outputs a stream of feature data which contains both original feature instances and any new instances, when applicable, which conform to new hybrid feature types that are defined in a Hybrid Feature Catalogue. The selection process is shown in Figure 16-1, and consists only of selecting the applicable predefined combination (PDC) that corresponds to the user or system settings and which lists exactly the currently loaded products. There should be only one applicable PDC after these steps; however, if there is a tie it must be broken by external tie-breaking methods such as allowing the user to select a combination based on the use conditions and description attributes of the PDC.



Figure 16-1 - Process for selecting rules

Product combinations are lists of data products. When associated with a ruleset or an individual rule, it means that the ruleset or single rule applies to the combinations of the data products listed. Each product combination may be assigned an interoperability level, if implemented, and descriptive attributes which indicate its purpose and applicability to the human end-user and catalogue developer. Levels are used to break up the functionality of the interoperability catalogue model into logical modules as appropriate for the use case. If the use case does not call for breaking up the functionality of the interoperability catalogue model only a single level 1 is required.

Display planes act as a means of layering features in the end-user's display. Each feature in the output stream is assigned to a display plane. Each plane is assigned a display order relative to other planes. The Interoperability Catalogue uses display planes for interleaving features from different products, with features in higher priority planes overlying those in lower priority planes wherever they overlap. In the case of coincident or overlapping symbols, the implementer may use appropriate methods to avoid displaying partial symbols, or "grafting" part of lower-layer symbols onto symbols in upper layers.

The Interoperability Catalogue allows suppression of feature classes where the presence in one product of features of a given feature type causes the removal of all features of a specified feature type from another specified data product. This is supposed to be used when the two feature types represent the same kind of data but one data product is preferred over the other. For example, level of detail in the preferred data product, additional feature characteristics in the preferred product, etc.

The Interoperability Catalogue also allows suppression of feature instances whereby only feature instances meeting specified conditions are suppressed in favour of feature instances from a different product. The conditions are described in terms of specific characteristics such as attribute values or combinations of values of different attributes, or the use of specific classes of spatial primitives such as all point features of the specified type.

EXAMPLE: Restricted area features from type A datasets with category attribute = (nature reserve), (bird sanctuary), (seal sanctuary), (ecological reserve), or (coral sanctuary) are suppressed in favour of restricted area features from type B datasets due to more details.

Advanced interoperability functionality includes hybridization of features. Hybridization consists of combining feature data from different products in the input stream into a new type in the output stream – new in the sense that the output feature type is not defined in any feature catalogue of the input products. Such combined types are intended for producing resultant features with enhanced characteristics, for example by enhancing the attribute set from one product with additional attributes derived from another data product.

EXAMPLE: water level information from one product is combined with bathymetry in another and with high definition bathymetry in a third product to create go and no-go areas that increase and decrease with changing water level.

The hybrid feature is defined in a hybrid feature catalogue (must comply with Part 5) and its portrayal is defined in a hybrid portrayal catalogue (must comply with Part 9) that is bundled with the interoperability catalogue in an exchange set.

16-4.3 Overview of processing

Interoperability processing can either precede or follow portrayal processing (except rendering, which converts feature data into graphics and is necessarily the step just before actual display). A mixed processing model, where interoperability processing is done both before and after portrayal processing, is also possible.

- Interoperability before regular portrayal processing: Feature data from S-100-based datasets is an
 input to the interoperability processor, along with the Interoperability Catalogue and context
 parameters. The interoperability processor filters and interleaves feature data according to the
 Interoperability Catalogue and interoperability level selected by the user and passes the resultant
 feature data to the portrayal processor, which uses the Portrayal Catalogue for individual products
 to generate drawing instructions for the display processor.
- Interoperability after regular portrayal processing: Feature data from S-100-based datasets flows to the portrayal processor. The portrayal processor transforms them into drawing instructions. The drawing instructions flow to the interoperability processor. The interoperability processor filters and interleaves the drawing instructions according to the Interoperability Catalogue and interoperability level selected by the user and passes the resultant drawing instructions to the display processor.

The two processing options described above are shown in Figure 16-2.



Figure 16-2 - Basic interoperability processing

In any level of processing except Level 0 (interoperability off), data products to be loaded are selected by the system according to the list in the predefined combination selected by the user selection from among those listed in the catalogue.

Feature data from products not listed in the Interoperability Catalogue are passed through to portrayal processing as described in S-100 Part 9 (stage *Portrayal Processing*) without any intermediate stages in interoperability processing, and displayed by ordinary S-100 portrayal processing according to their individual Portrayal Catalogues.

16-4.4 Interoperability Catalogue Data Model

16-4.4.1 Introduction

The Interoperability Catalogue specifies the relative display prioritization of feature types and feature instances, as defined in individual Product Specifications, in relation to other feature types and feature instances, which may be defined in any of the data products declared to be within the scope of the Interoperability Catalogue.

An Interoperability Catalogue describes display planes, predefined combinations, feature instance and feature layer suppression rules.

The interoperability Catalogue (IC) utilizes the ISO **CT_Catalogue** class defined in ISO 19139 (implemented in ISO 19115-3) as a super-type for header information. The body of the Interoperability Catalogue consists of subsections encoding the rules for display planes, feature priorities, feature interleaving, and available predefined combinations:

- display planes, indicating order of planes, viewing group, and drawing priority;
- predefined combinations and operations on feature types or feature instances for each combination;

An Interoperability Catalogue must be an XML document which conforms to the Interoperability Catalogue Schema which can be downloaded from the IHO website. Figure 16-3 shows the Interoperability Catalogue model.





16-4.4.2 Interoperation Conceptual Types

The following clauses describe the different conceptual elements that may be used in an interoperability catalogue.

16-4.4.2.1 S100_IC_InteroperabilityCatalogue

An interoperability catalogue contains operations and rules for the interoperation of a set of S-100-based data products.

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|-----------------------------------|--|------|-----------------|---|
| Class | S100_IC_InteroperabilityCatalogue | Class that holds header information about an interoperability catalogue | | | Specialization of CT_Catalogue (ISO 19115-3) |
| Attribute | description | Description of the catalogue | 1 | CharacterString | |
| Attribute | comment | Any additional comments | 01 | CharacterString | |
| Attribute | interoperabilityLevel | The highest level of interoperability functionality encoded within an instance of this type | 01 | Integer | |
| Attribute | requirementType | The type of authority or requestor responsible for the specifications, rules, or requirements based on which this catalogue was prepared | 1 | Enumeration | |
| Attribute | requirementDescription | Description of the source of the requirements or specifications upon which this catalogue is based. This might be the name of the country, company, OEM, port, pilot, etc | 1 | CharacterString | |
| Attribute | productCovered | The products covered by this catalogue | 2* | dataProduct | |
| Attribute | name | The name for the catalogue | 1 | CharacterString | Inherited from CT_Catalogue |
| Attribute | scope | Subject domain of the catalogue | 1* | CharacterString | Inherited from CT_Catalogue |
| Attribute | fieldOfApplication | Description of the use to which this catalogue may be put | 0* | CharacterString | Inherited from CT_Catalogue |
| Attribute | versionNumber | The version number of the Product Specification | 1 | CharacterString | Inherited from CT_Catalogue |

| Role Name | Name | Description | Mult | Туре | Remarks |
|-------------|-------------------------------|---|------|--|--|
| Attribute | versionDate | The version date of the Product Specification | 1 | Date | Inherited from CT_Catalogue |
| Attribute | language | The language used for this catalogue | 01 | CharacterString | Inherited from CT_Catalogue |
| Attribute | locale | provides information about alternatively used localized character strings | 01 | PT_Locale (ISO 19115-1) | Inherited from CT_Catalogue |
| Attribute | characterSet | Character set used in the catalogue | 01 | MD_CharacterSetCode (ISO 19115-1) | Inherited from CT_Catalogue must have value=utf8 |
| Composition | displayPlanes | Container for one or more S100_IC_DisplayPlane elements | 1* | <sequence>S100_IC_DisplayPlane</sequence> | Ordered list of one or more S100_IC_DisplayPlane elements |
| Composition | predefinedProductCombinations | Container for predefined product combinations and the interoperability operations for each | 0* | <sequence>S100_IC_PredefinedCombination</sequence> | Sequence of S100_IC_PredefinedCombination elements |
| Composition | hybridizationRules | Container for hybridization rules | 0* | <sequence>S100_IC_HybridizationRule</sequence> | |
| Role | hybridFC | Reference to hybrid feature catalogue used by operations in this Interoperability Catalogue | 0* | CharacterString | |
| Role | hybridPC | Reference to hybrid portrayal catalogue used by operations in this Interoperability Catalogue | 0* | CharacterString | |

16-4.4.2.2 S100_IC_DisplayPlane

A display plane element acts as a container for display information for specified feature classes. The display order for the plane as a whole is provided in the S100_IC_DisplayPlane element. All the types within an instance of S100_IC_DisplayPlane have the same display order (encoded in attribute order) relative to feature types in another instance of S100_IC_DisplayPlane. Instances of display plane can be characterized by interoperability level, which allows the encoding of different sets of operations depending on how tightly integrated the user desires the products to be on the resultant display.

Assigning feature types to display planes enables the interleaving of feature layers during portrayal by indicating the display plane, priority, and rendering order of the types assigned to a display plane. **S100_IC_DisplayPlane** assigns subsets of feature types to display planes and defines the viewing group, drawing priority, and significance for each feature type in the plane. An **S100_IC_DisplayPlane** element may include more than one feature type.

A feature type may be referenced in more than one **S100_IC_DisplayPlane**, but the entries in different display planes must be distinguished by different attribute-value combinations or spatial primitives so that the actual instances of features are partitioned unambiguously between different display planes.

The portrayal of feature types not mentioned in any S100_IC_DisplayPlane component is undefined until ordinary portrayal processing.

An instance of S100_IC_DisplayPlane must contain at least one instance of S100_IC_Feature or S100_IC_DrawingInstruction. An instance of S100_IC_DisplayPlane may contain both S100_IC_Feature and S100_IC_DrawingInstruction, subject to the constraint below.

For **S100_IC_Feature** and **S100_IC_DrawingInstruction** with the same combination of [featureCode, product, geometryType, and attributeCombination] and in the same **S100_IC_DisplayPlane** container:

- S100_IC_Feature. drawingPriority and S100_IC_DrawingInstruction. drawingPriority must have the same values.
- **S100_IC_Feature.viewingGroup** and **S100_IC_DrawingInstruction.viewingGroup** must have the same values.

| Role Name | Name | Description | Mult | Туре | Remarks |
|-------------|-----------------------|--|------|---|---|
| Class | S100_IC_DisplayPlane | Each display plane identifies all features and their drawing priority within the plane. The order in which display planes are rendered is also given. | | | Composition component of S100_IC_InteroperabilityCatalogue, container displayPlanes |
| Attribute | identifier | Unique identifier of the display plane | 1 | CharacterString | Must be unique |
| Attribute | name | Name of display plane | 1 | CharacterString | Under radar, over radar, etc. |
| Attribute | order | Used to sort the rendering order of display planes. Display planes with larger values are drawn above those with lower values. | 1 | Integer | Refer Part 2 clause 2b-4.2.22 and S- 100 Part 9, clauses 9-11.1.4 & 9-12.1 Positive: Above RADAR Zero: Reserved for RADAR Negative: Below RADAR |
| Attribute | description | description of the display plane | 1 | CharacterString | |
| Attribute | interoperabilityLevel | The highest level of interoperability functionality encoded within an instance of this type | 01 | Integer | |
| Composition | features | Container for S100_IC_Feature elements | 0* | <sequence>S100_IC_Feature</sequence> | At least one S100_IC_Feature or S100_IC_DrawingInstruction element must be included in a display plane element |
| Composition | drawingInstructions | Container for S100_IC_DrawingInstruction elements | 0* | <sequence>S100_IC_DrawingInstruction</sequence> | At least one S100_IC_Feature or S100_IC_DrawingInstruction element must be included in a display plane element |

Table 16-2 S100_IC_DisplayPlane

16-4.4.2.3 S100_IC_Feature

The **S100_IC_Feature** element describes the display parameters for all features of a specific feature type in a specific product. The **S100_IC_Feature** element determines the order of drawing the feature type identified by its featureCode attribute relative to other feature types in the same display plane. It also specifies the viewing group to which the feature is assigned. Its applicability can be optionally restricted to a subset of instances of the feature type by additional attributes that specify the type of spatial primitive and indicate specific values of thematic attributes.

The **S100_IC_Feature** element in Interoperability Catalogues is similar in operation to the layering and priority aspects of the **DrawingInstruction** element in Portrayal Catalogues (see S-100 Part 9 - Portrayal), and therefore has attributes that are equivalent to some of the attributes and roles of the Portrayal Catalogue element. Where there is an exact correspondence with a portrayal catalogue element, the element in the Interoperability Catalogue element. The correspondences are summarized in Table 16-19 at clause 16-5.

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|----------------------|---|------|------------------------------|-------------------------------------|
| Class | S100_IC_Feature | Information that guides the relative layering and drawing priority of feature types during portrayal | | | |
| Attribute | identifier | Internal identifier of the catalogue element | 1 | CharacterString | |
| Attribute | featureCode | The code assigned to the feature type in feature catalogue for the product indicated in the product attribute | 1 | CharacterString | |
| Attribute | product | A data product | 1 | dataProduct | |
| Attribute | geometryType | The type of spatial primitive that indicates the location | 0* | S100_FC_SpatialPrimitiveType | |
| Attribute | attributeCombination | Describes attribute-value filters to be applied to the specified features | 0* | CharacterString | See Clause 16-4.2.3 |
| Attribute | drawingPriority | Drawing priority of feature type in the display plane | 1 | Integer | Refer S-100 Part 9, clause 9-11.2.2 |
| Attribute | viewingGroup | The viewing group of the feature type | 1 | Integer | Refer S-100 Part 9, clause 9-12.1 |

Table 16-3 S100_IC_Feature

NOTE: **S100_IC_Feature** and **S100_IC_DrawingInstruction** elements operate in essentially the same way as far as assignment of drawing priority, and display planes is concerned. They differ in that **S100_IC_DrawingInstruction** provides an optional attribute to substitute the symbolization elements of the drawing instruction. **S100_IC_Feature** should be used for Interoperability Catalogues that are designed for systems where interoperability processing precedes the generation of drawing instructions. **S100_IC_DrawingInstruction** should be used for Interoperability Catalogues that are designed for systems where interoperability processing precedes the generation of drawing precedes the generation of drawing instructions. It should also be used in all catalogues where substitution of symbolization is necessary.

16-4.4.2.4 S100_IC_DrawingInstruction

Drawing instructions specify the display order used by the rendering engine in producing the portrayal output of a given feature type/geometric primitive type/attribute value combination. The **S100_IC_DrawingInstruction** element determines the order of drawing the feature type identified by its **featureCode** attribute relative to other feature types in the same display plane. The applicability of an **S100_IC_DrawingInstruction** to feature types can be further restricted by the type of spatial primitive and values of thematic attribute, using **geometryType** and **attributeCombination** attributes of the **S100_IC_DrawingInstruction** class.

The **S100_IC_DrawingInstruction** element in Interoperability Catalogues is similar in operation to the layering and priority aspects of the **DrawingInstruction** element in Portrayal Catalogues (see S-100 Part 9 - Portrayal), and therefore has attributes that are equivalent to some of the attributes and roles of the Portrayal Catalogue element. Where there is an exact correspondence with a Portrayal Catalogue element, the element in the Interoperability Catalogue element supersedes the Portrayal Catalogue element. The correspondences are summarized in Table 16-19 in clause 16-5. (Definitions of the Portrayal Catalogue attributes are provided in S-100 Part 9, clause 9-11.2.2).

The **S100_IC_DrawingInstruction** element contains the **substituteSymbolization** attribute that allows substitution of symbolization instructions generated by portrayal processing. Note that the display instruction XML elements defined in the presentation XML schema S-100 Part 9 (**pointInstruction**, **lineInstruction**, etc.) cannot be used directly because, being extensions of the base type **DrawingInstruction** in that schema, they: (a) reference individual feature and spatial instances, and (b) contain viewing group, display plane, and drawing priority as mandatory elements, which would be redundant.

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|----------------------------|---|------|------------------------------|---|
| Class | S100_IC_DrawingInstruction | Information that guides the relative layering and drawing priority of drawing instruction during portrayal. | | | |
| Attribute | identifier | Internal identifier of the instruction group | 1 | CharacterString | |
| Attribute | featureCode | The code assigned to the feature type in feature catalogue for the product indicated in the product attribute | 1 | CharacterString | Corresponds to the feature reference for drawing instructions in S-100 Part 9 |
| Attribute | product | A data product | 1 | dataProduct | |
| Attribute | geometryType | The type of spatial primitive that indicates the location | 0* | S100_FC_SpatialPrimitiveType | |
| Attribute | attributeCombination | Describes attribute-value filters to be applied to the specified features | 0* | CharacterString | See Clause 16-4.2.3 |
| Attribute | drawingPriority | The drawing priority of the group | 1 | Integer | Refer S-100 Part 9, clause 9-11.2.2 |
| Attribute | viewingGroup | The viewing group of the feature type | 1 | Integer | Refer S-100 Part 9, clause 9-12.1 |

Table 16-4 S100_IC_DrawingInstruction

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|-------------------------|--|------|-----------------|--|
| Attribute | substituteSymbolization | Substitute for the symbolization content of drawing instructions. This can be any element of the drawing instruction not defined in the abstract class DrawingInstruction defined in S-100 Part 9, clause 9-11.2, but defined in the relevant descendant of that class | 01 | CharacterString | The string must consist of one or more XML fragments constructed according to the Presentation schema in S-100 Part 9 or the equivalent in a non-XML syntax. A CDATA section may be used to avoid the explicit encoding of character entities for special characters |

NOTE: Even if the Presentation schema in S-100 Part 9 is used, specific code may need to be provided to validate the content of the **substituteSymbolization** attribute instead of depending on purely XML schema validation. The content of this attribute is not prescribed by this specification and may be a fragment of XML, or interpretable code or rules, etc., in a non-XML syntax. It may be enclosed in a <! [CDATA[...]] > section so that XML validators treat it as character data instead of XML. If the content is XML, it must be well-formed (for example have balanced opening and closing tags).

16-4.4.2.5 S100_IC_SuppressedFeatureLayer

Each instance of this element identifies a feature type in a specific data product.

| Role Name | Name | Description | Mult | Туре | Remarks |
|--------------|--------------------------------|--|------|---|---|
| Class | S100_IC_SuppressedFeatureLayer | Describes operations for suppressing all instances of a feature type in one product by features from another product | | | |
| Attribute | identifier | Internal identifier of the catalogue element | 1 | CharacterString | |
| Attribute | featureCode | Feature type code in the FC for the product mentioned in attribute product | 1 | CharacterString | |
| Attribute | product | The data product for the type being replaced | 1 | dataProduct | |
| Role | featureRef | References to replacement features' display specifications in the display planes section of the Interoperability Catalogue | 0* | <reference>S100_IC_Feature</reference> | Replacement by multiple feature types is intended for associated feature types, for example different feature types that make up a traffic separation scheme. |
| Role | drawingInstructionRef | Reference to S100_DrawingInstruction element | 0* | <reference>S100_IC_DrawingInstruction</reference> | |

Table 16-5 S100_IC_SuppressedFeatureLayer

NOTE: Only one of featureRef or drawingInstructionRef can be used in an instance.

NOTE: If both feature and drawing instruction references are empty, the type is suppressed without being replaced.

16-4.4.2.6 S100_IC_PredefinedCombination

A predefined combination element defines a collection of data products for which a common set of interoperability operations have been defined in the Interoperability Catalogue. Instances of predefined combinations can be characterized by interoperability level, which allows the segmentation of different sets of interoperability operations depending on how tightly integrated the user desires the products to be on the resultant display, see Clause 16-8 for more details.

Predefined combination element can specify the following types of interaction between its listed products.

- Operations on selected instances of a feature type or conversion of input feature data into new feature data involving only thematic attributes.
- Operations involving operations on spatial attributes and possibly thematic attributes as well.

The simplest operations on instances are replacement of selected instances from one product by selected instances from another product. These are described by associated **S100_IC_SuppressedFeatureInstance** elements. More complex operations, including conversion of input feature instances into new features (hybridization) are described by associated **S100_IC_HybridFeature** elements. The replacement and hybridization rules are described in Clause 16-6.

Predefined combinations can be linked to **S100_IC_DisplayPlane** elements by means of references in the **S100_IC_PredefinedCombination** elements.

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|-------------------------------|---|------|-----------------|--|
| Class | S100_IC_PredefinedCombination | Pre-defined combinations are identifiable pre-set collections of recommended and optional S-NNN data products which are expected to be loaded by the user under specific conditions or for specified tasks. Each pre-defined combination is basically a package of data products, display priorities, context parameters, user settings, portrayal catalogues, etc. | | | Composition component of S100_IC_InteroperabilityCatalogue |
| Attribute | identifier | Identifier of the predefined combination | 1 | CharacterString | For example, a sequence number, UUID or URN unique to the PDC in the catalogue. May be globally unique, but must be unique within the catalogue at least |
| Attribute | name | Name of combination | 1 | CharacterString | |
| Attribute | description | Brief description of combination | 1 | CharacterString | |
| Attribute | useConditions | Conditions for which the combination is designed | 1 | CharacterString | |

Table 16-6 S100_IC_PredefinedCombination

| Role Name | Name | Description | Mult | Туре | Remarks |
|-------------|-------------------------|---|------|---|---|
| Attribute | interoperabilityLevel | The highest level of interoperability functionality encoded within an instance of this type | 01 | Integer | |
| Attribute | includedProduct | Products loaded in this combination and referenced by operations and rules that apply to this combination | 2* | dataProduct | A combination must use at least 2 data products |
| Role | displayPlaneRef | Reference to an S100_IC_DisplayPlane element in this Interoperability Catalogue | 0* | <reference>S100_IC_DisplayPlane</reference> | |
| Composition | derivedFeatures | Container for S100_IC_SuppressedFeatureInstance or S100_IC_HybridFeature elements (concrete specializations of S100_IC_FeatureDerivation) | 0* | <sequence> of sub-classes of S100_IC_FeatureDerivation</sequence> | |
| Composition | suppressedFeatureLayers | Container for S100_IC_SuppressedFeatureLayer | 0* | <sequence> S100_IC_SuppressedFeatureLayer</sequence> | |

NOTE: A system can allow the user to initiate the loading of multiple data products and activate multiple parameter settings as a single action, by selecting from a list of pre-defined combinations, instead of loading and unloading individual data products.

16-4.2.2.7 S100_IC_FeatureDerivation

S100_IC_FeatureDerivation is an abstract super-class for different types of feature hybridization operations. Individual primary and secondary inputs are suppressed from being rendered and only the resulting derived feature is added to the data stack.

The resulting derived feature does not need to have any hybrid characteristics, that is, one restricted area replaced with another restricted area will use the regular PC/FC of the primary product. However, if the result feature needs to be supported by any custom FC or PC elements, they must be defined under hybrid FC and hybrid PC accordingly.

A rule for creating the feature must be described in the rules section

| Table 16- | 7 S100 | _IC_ | FeatureDerivation |
|-----------|--------|------|-------------------|
|-----------|--------|------|-------------------|

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|---------------------------|---|------|-----------------|-----------------|
| Class | S100_IC_FeatureDerivation | Derived features are created by consolidating features from 2 or more different products into one final view, so the changes can include geometry, attribution and/or portrayal (depending on the interoperability level) | | | Abstract class. |
| Attribute | identifier | Internal identifier of the catalogue element | 1 | CharacterString | |
| Attribute | primaryProduct | One of the two interoperating data products | 1 | dataProduct | |

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|----------------------|---|------|--|--|
| Attribute | primaryFeatureCode | Feature type code in the FC for the product mentioned in primaryProduct | 1 | CharacterString | |
| Attribute | primarySelector | Selection expression for instances of the first feature type | 01 | FeatureSelector | If omitted, all instances of the type are included. Example 1: categoryOfObstruction = 5 Example 2: waterLevelEffect = 4 AND WITHIN(<primary>, <secondary>)</secondary></primary> |
| Attribute | secondaryProduct | The other interoperating data product | 1 | dataProduct | |
| Attribute | secondaryFeatureCode | Feature type code in the FC for the product mentioned in secondaryProduct | 1 | CharacterString | |
| Attribute | secondarySelector | Selection expression for instances of the second feature type | 01 | FeatureSelector | If omitted, all instances of the type are included. Example 1: expositionOfSounding = 3 Example 2: categoryOfPile = 2 AND WITHIN(<primary>, <secondary>)</secondary></primary> |
| Attribute | outputProduct | Data product of the resulting hybrid feature | 1 | dataProduct | Default value = HYBRID, to indicate the result is a hybrid feature. This is a default – it can be set to other allowed values from the dictionary in specific cases. |
| Attribute | outputFeatureCode | Feature type code in the hybrid FC | 1 | CharacterString | |
| Role | featureRef | Reference to the output feature's display specification in the display planes section of the interoperability catalogue | 1 | <reference>S100_IC_Feature</reference> | |

16-4.2.2.8 S100_IC_SuppressedFeatureInstance

S100_IC_SuppressedFeatureInstance is a class for suppressing feature instances according to attribute combinations in one product with features instances in another product. The primary product attribute combination specifies the feature instance(s) that will be replaced. The secondary product combination is the product to replace the suppressed instance(s) and should be identical to the output product attribute combination. If two different set of features instances are to be suppressed by one common set of feature instances, two instances of S100_IC_SuppressedFeatureInstance are needed.

Table 16-8 S100_IC_SuppressedFeatureInstance

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|-----------------------------------|---|------|---|---|
| Class | S100_IC_SuppressedFeatureInstance | Operations for replacement of feature instances in one product by instances in another product. | | | Sub-class of S100_IC_FeatureDerivation The "secondary" product replaces the "primary". |
| Role | creationRule | Reference to a rule defined in the hybridization rules section of the catalogue | 01 | <reference>S100_IC_SimpleRule</reference> | |

NOTE: If attributes bindings of the output are the same as secondary product type, the **outputProduct** and **outputFeatureCode** should be the same as the **secondaryProduct** and **secondaryFeatureCode**. If the attribute bindings change in a way that is incompatible with the feature catalogue for the secondary product, **S100_IC_HybridFeature** must be used instead.

16-4.2.2.9 S100_IC_HybridFeature

S100_IC_HybridFeature is a class for selecting primary and secondary inputs that will be suppressed from being rendered and replaced by a feature derived from the inputs.

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|-----------------------|---|------|---|--|
| Class | S100_IC_HybridFeature | Class used to create a feature by combining feature types from two or more products for the purposes of an interoperable display | | S100_IC_FeatureDerivation | Sub-class of S100_IC_FeatureDerivation |
| Role | creationRule | Reference to a rule defined in the hybridization rules section of the catalogue | 01 | <reference>S100_IC_CompleteRule</reference> | |
| Role | creationRule | Reference to a rule defined in the hybridization rules section of the catalogue | 01 | <reference>S100_IC_ThematicRule</reference> | |

Table 16-9 S100_IC_HybridFeature

NOTE: Only one creationRule can be used in an instance.

16-4.2.2.10 S100_IC_HybridFeatureCreationRule

S100_IC_HybridFeatureCreationRule is an abstract super-class for different types of hybridization rules. *This functionality needs to be worked out but OGC Filter seems to be the ideal option for defining data filtering logic.* Overall, the output from execution of S100_IC_HybridFeatureCreationRule is a set of hybrid features for which predefined FC, PC and display plane definitions already exist so such feature will be suitable for passing to the portrayal engine for processing just like any other S-100 features. Instances of S100_IC_HybridFeatureCreationRule can be characterized by interoperability level, which allows the segmentation of different sets of interoperability apprations depending on how tightly integrated the user desires the products to be on the resultant display. See 16.8 for more

interoperability operations depending on how tightly integrated the user desires the products to be on the resultant display, see 16-8 for more details.

Table 16-10 S100_IC_HybridFeatureCreationRule

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|-----------------------------------|---|------|-----------------|--|
| Class | S100_IC_HybridFeatureCreationRule | Class to capture hybrid feature creation rule captures the entire data filtering logic (that is, finding all features to be operated on) as well as the entire processing logic. | | | Abstract class |
| Attribute | interoperabilityLevel | The highest level of interoperability functionality encoded within an instance of this type | 01 | Integer | |
| Attribute | ruleldentifier | Rule identifier | 1 | CharacterString | Mandatory unique ID used for references |

16-4.2.2.11 S100_IC_CompleteRule

This class is used for rules describing how a set of feature instances is combined to create a hybrid feature type. Complete rules may operate on both thematic and spatial attributes

Table 16-11 S100_IC_CompleteRule

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|----------------------|--|------|------|--|
| Class | S100_IC_CompleteRule | Class to capture rule describing how a set of feature instances is combined to create a hybrid feature type. | | | Subclass of S100_IC_HybridFeatureCreationRule |

16-4.2.2.12 S100_IC_ThematicRule

This class is used for rules describing how a set of feature instances is combined to create a hybrid feature type. Thematic rules may operate on only thematic attributes. The input features are required to have spatially equal geometry within a tolerance set by the system.

Table 16-12 S100_IC_ThematicRule

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|----------------------|---|------|------|---|
| Class | S100_IC_ThematicRule | Rule describing how a set of feature instances is combined to create a hybrid feature type. | | - | Subclass of S100_IC_HybridFeatureCreationRule |

16-4.2.2.13 S100_IC_SimpleRule

This class is used for rules describing how a set of feature instances is combined to create a hybrid feature type. Simple rules may operate only on thematic attributes. Simple rules treat thematic attributes uniformly during hybridization and therefore do not mention specific attributes. Location/extent spatial attributes of all input features must be spatially equal.

Table 16-13 S100_IC_SimpleRule

| Role Name | Name | Description | Mult | Туре | Remarks |
|-----------|--------------------|--|------|------|---|
| Class | S100_IC_SimpleRule | Rule describing how a set of feature instances is combined to create a hybrid feature type | | | Subclass of S100_IC_HybridFeatureCreationRule |

16-4.2.2.14 S100_IC_HybridFC

Class for holding reference to a Feature Catalogue defining any "hybrid feature types" that are created by combining feature types from two or more products for the purposes of an interoperable display.

Table 16-14 S100_IC_HybridFC

| Role Name | Name | Description | Mult | Туре | Remarks |
|-------------------------|--------------------|--|------|--------------------------|--|
| Class | S100_IC_HybridFC | Feature Catalogue defining any "hybrid feature types" that are created by combining feature types from two or more products for the purposes of an interoperable display. | | S100_FC_FeatureCatalogue | The Interoperability Catalogue contains references to local resources (files) containing hybrid Feature Catalogues |
| Attributes and Roles | (See S-100 Part 5) | | | | |

16-4.2.2.15 S100_IC_HybridPC

Class for holding reference to a Portrayal Catalogue defining portrayal rules for the "hybrid feature types" defined in a Hybrid Features Catalogue.

Table 16-15 S100_IC_HybridPC

| Role Name | Name | Description | Mult | Туре | Remarks |
|-------------------------|--------------------|---|------|---------------------------------|--|
| Class | S100_IC_HybridPC | Portrayal catalogue defining portrayal rules for the "hybrid feature types" defined in a Hybrid Features Catalogue. | | (S-100 Part 9) PortrayalCatalog | The Interoperability Catalogue contains references to local resources (files or folders) defining hybrid PCs |
| Attributes and Roles | (See S-100 Part 9) | | | | |

16-4.2.2.16 FeatureSelector

A data type for holding a template, logical expression, or match condition that, given a feature instance as parameter, can be evaluated to produce a TRUE/FALSE result.

Table 16-16 FeatureSelector

| Туре | Name | Description | Derivation | Remarks |
|----------|-----------------|---|----------------------------|-------------------------------|
| datatype | FeatureSelector | A template, logical expression, or match condition that, given a feature instance as parameter, can be evaluated to produce a TRUE/FALSE result | subtype of CharacterString | Example: XSLT match condition |

NOTE: Format and expression language must be defined in the implementation.

16-4.2.2.17 requirementType

A code list for sources of the interoperability catalogue or the person or party according to whose recommendations the catalogue was prepared.

| Role Name | Name | Description | Code | Remarks |
|---------------|-----------------|---|------|---------|
| S100_Codelist | requirementType | The source of the catalogue or the person or party according to whose recommendations the catalogue was prepared | | |
| Literal | IHO | Original IHO Interoperability Catalogue | 1 | |
| Literal | OEM | Prepared according to requirements specified by OEM or systems integrator | 2 | |
| Literal | national | Prepared according to requirements specified by a national government, group of national governments (for example the European Union), or governmental agency such as a national shipping authority or the Coast Guard. | 3 | |
| Literal | local | Prepared according to requirements specified by a sub-national governmental authority such as a state, province, or county | 4 | |
| Literal | port | Prepared according to requirements specified by a harbormaster's office or port authority | 5 | |
| Literal | company | Prepared according to requirements specified by the owner, charterer, or operator | 6 | |
| Literal | pilot | Prepared according to requirements specified by the vessel's master | 7 | |

Table 16-17 requirementType

| Literal | master | Prepared according to requirements specified by a pilot | 8 | |
|---------|--------|---|---|--|
| Literal | other | Other source | 9 | |

Codelist Type: open enumeration

Encoding for extra values: other: <CharacterString> (Format of <CharacterString>: [a-zA-Z0-9]+([a-zA-Z0-9]+)* - See S-100 Part 3, clause 3-6.7).

16-4.4.2.18 dataProduct

A closed dictionary codelist of S-100 based products.

The allowed values are defined in the dictionary file, which is a component of individual interoperability specifications implementing this Part. If any operations or rules in an Interoperability Catalogue produce a hybrid feature, one of the entries in the dictionary must correspond to a notional "HYBRID" data product for such features.

Table 16-18 dataProduct

| Role Name | Name | Description | Code | Remarks |
|---------------|---------------------------------|-----------------------|------|---|
| S100_Codelist | dataProduct | List of data products | | Data products conforming to the Specification identified by the item name, in the IHO list of S-100 based Product Specifications. This is a closed dictionary codelist (see S-100 1-4.8, 3-5.3.11, 3-6.7). |
| Literal(s) | (see individual specifications) | | | |

Codelist Type: closed dictionary, URI format: urn:mrn:...

URI tags must be fully specified in the implementation specification. E.g. urn:mrn:iho:prod:s98:1:0:0:products for a master list of products that is specified in S-98 as being covered by S-98 Version 1.0.0 interoperability catalogues.

The dictionary format used in interoperability catalogues is the ISO 19115-3 "codelist catalogue" format. An example using this format is depicted in Figure 16.4. The whole data products codelist is contained in the $CT_Codelist$ XML element, identified by the XML ID "urn.mrn.iho.prod.s98.1.0.0.products" (the *id* attribute of $CT_Codelist$). Individual data products are listed in *codeEntry* elements within the $CT_Codelist$, identified by their own *id* attributes. In the example, S-101 is identified by the XML ID "s101" in the expanded *codeEntry* element. Note that the ":" characters in the codelist URI are replaced by "." in the corresponding XML id elements, due to XML syntax constraints. The combination of *id* values of $CT_Codelist$ and $CT_CodelistValue$ elements is sufficient to identify a data product within a given dictionary file. The name (or other identifier) of the dictionary file must be specified in the individual interoperability specification. The product specification must also describe how dictionary files are made available to individual end user systems.

<cat:codelistItem>

- <cat:CT Codelist id="urn.mrn.iho.prod.s98.1.0.0.products">

<cat:identifier>...</cat:identifier>

<cat:name>...</cat:name>

<cat:definition>...</cat:definition>

<cat:description>...</cat:description>

-<cat:codeEntry>

-<cat:CT_CodelistValue id="s101">

-<cat:identifier>

<gco:ScopedName codeSpace="http://www.iho.int/S100/products">S-101</gco:ScopedName> </cat:identifier>

-<cat:name>

<gco:ScopedName codeSpace="http://www.iho.int/S100/products">S-101 ENC</gco:ScopedName>

- </cat:name>
- -<cat:definition>

<gco:CharacterString>S-101 Electronic Navigation Charts</gco:CharacterString>

</cat:definition>

-<description>

<gco:CharacterString>ENC data product</gco:CharacterString>

</description>

</cat:CT_CodelistValue>

</cat:codeEntry>

<cat:codeEntry>...</cat:codeEntry>

<cat:codeEntry>...</cat:codeEntry>

<cat:codeEntry>...</cat:codeEntry>

<cat:codeEntry>...</cat:codeEntry>

<cat:codeEntry>...</cat:codeEntry>

</cat:CT_Codelist>

</cat:codelistItem>

Figure 16-4 - Example of data products dictionary entry

16-4.2.3 Filters

Attribute-value combination filters (the **attributeCombination** attribute of S100_IC_Feature) are strings of the form <a trial combination attribute of S100_IC_Feature), where:

- a) <attr> is the camel case code of the attribute;
- b) <op> is one of "=", "!=", "in", "notIn", "gt", "ge", "It", "le", "null";
- c) <value> is a decimal number, integer, numeric code, or string, or a list of values. Strings must be enclosed in double quotes: "" with embedded double-quotes or \ characters preceded by a \ character.

The <attr>, <op>, and <value> components are separated by blank or tab characters¹.

Sub-attributes of complex attributes can be indicated in <attr> fields using a restricted subset of relative path expressions as specified in the W3C XPath specification (§ 3.3.1 XML Path Language (XPath) 3.1). The restrictions are:

- Paths are relative to the individual feature as the context node.
- Only the "child" axis is permitted and the optional "child::" prefix is not used.
- Predicates as described in the XPath specification are not used.

The effect is to allow <attr> fields to describe sub-attributes in terms of camel case codes separated by "/" characters. (It also allows a simple attribute to be designated by its camel case code alone as described above.)

EXAMPLE 1: An attributeCombination with value *categoryOfRadioStation* = 20 selects features with **categoryOfRadioStation** attributes that have the value 20 (AIS Base station).

EXAMPLE 2: An **attributeCombination** with value *featureName/language* = *"eng"* selects features with **featureName** attributes that have a language sub-attribute having the value "eng".

Note (informative): Selectors may relax these restrictions and use a larger subset of XPath. Details should be elaborated in the implementation specification.

16-5 Correspondence to and supersession of portrayal catalogue elements

Correspondences between Interoperability Catalogue and Portrayal Catalogue elements are summarized in Table 16-19 below. In all cases, the supersession of Portrayal Catalogue display by Interoperability Catalogue display applies only to the subset of features remaining after applying the filter described by the attributes geometryType and attributeCombination.

| Table 16-19 - Correspondences between display instruction elements in Interoperability Catalogues and |
|---|
| Portrayal Catalogues |

| Interoperability element | Catalogue | Portrayal Catalogue element | Interoperability Catalogue supersedes Portrayal Catalogue? | Note |
|-----------------------------|-----------|---|--|---|
| product | | (implicit) aggregation in DisplayList | Not applicable | Implicit in aggregation in DisplayList; the product can be identified from data product to which the Portrayal Catalogue as a whole applies. |

¹ More expressive filter expressions can be developed if required for advanced interoperability.

| drawingPriority (S100_IC_DrawingInstruction) drawingPriority (S100_IC_Feature) | drawingPriority | Y | |
|---|--|--|--|
| identifier | | Not applicable | |
| viewingGroup | viewingGroup | Y | |
| geometryType | (implicit) | Y | Can be derived from the implementing element in the Portrayal Catalogue, for example PointInstruction, AreaInstruction, LineInstruction. |
| attributeCombination | (XSLT template) | Y | XSLT template in Portrayal Catalogue |
| featureCode | (implicit in XSLT template) | Not applicable (must be compatible by definition) | Interoperability Catalogue elements pertain to feature types or subsets of feature types; instances of drawing instructions in Portrayal Catalogues reference individual features. |
| (composition association) | displayPlane | Y | composition to S100_IC_DisplayPlane. |
| | scaleMinimum, scaleMaximum | Not applicable | |
| | featureReference association | Y | Interoperability Catalogue elements do not refer to individual instances (see Example 1). |
| | spatialReference association | Y | Interoperability Catalogue elements do not refer to individual instances (see Example 2). |
| substituteSymbolization (S100_IC_DrawingInstruction only) | (symbolization generated by portrayal processing) | Y | (See Example 3). |

EXAMPLE 1: A portrayal catalogue for an overlay product places all its features in the same display plane, but the interoperability catalogue splits them into over-radar and under-radar planes.

EXAMPLE 2: Feature **Current – Non-Gravitational** is allowed to have point, curve, or surface geometry. It is possible for a portrayal catalogue to place them in different viewing groups depending on the type of spatial primitive and for an interoperability catalogue to override that placement and put all instances of this feature, whether point, curve, or surface, in the same viewing group.

EXAMPLE 3: The area boundary symbolization is changed from a simple to a composite line style for area boundaries common to different types of area features, for example an anchorage area in product A bordering a marine protected area in product B.

16-6 Feature creation or replacement rule

Feature creation and replacement rules create new feature types by combining characteristics of specified feature types from the input data products. A feature creation/replacement rule basically transforms a collection of feature instances in the input stream into one or more different feature instances in the output stream. The created feature differs from all the input features, for example by adding properties of one feature to properties of another feature.

Since new feature types must be defined in a Feature Catalogue for ECDIS use, there is also a "hybrid" Feature Catalogue that contains feature type specifications for all the possible feature types which can be thus generated by rules in an Interoperability Catalogue. Similarly, the hybrid Portrayal Catalogue describes the portrayal of these feature types.

The process for applying such rules is illustrated in Figure 16-5 below.



Figure 16-5 - General process for feature creation and replacement rules

A hypothetical example of the operation of such a rule is depicted in Figure 16-6 below. On the left are two hypothetical input features:

- Feature instance DRGARE_12345 from a chart layer, an instance of feature type DredgedArea. Its geometry is an area, depicted alongside.
- Feature instance AISMessage_8472 from a met/hydro layer, an instance of feature type MetHydroDataAISMessage. Its geometry is a point, depicted alongside.

The operation of the rule results in two feature instances, described in Table 16-20 below:

| Output feature instance | Output feature Type | Defined in feature catalogue | Spatial attributes | Thematic attributes |
|-------------------------|------------------------|------------------------------------|---|--|
| RTWL_H01 | RealTimeWaterLevelArea | Hybrid FC IHOICFC01.XML | Circle centred at location of input AISMessage_8472 | Combination of DRGARE_12345 and AISMessage_8472 |
| DRGARE_H01 | DredgedArea | Chart FC | Spatial difference of original DRGARE_12345 and circle geometry of RTWLArea_H01 | Same as DRGARE_12345 |

 Table 16-20 - Features resulting from operation of hypothetical rule



Figure 16-6 - Illustrative example of a creation/replacement rule

16-7 Interoperability Catalogue Data Model and the General Feature Model

This clause elaborates on how General Feature Model (Part 3) elements are managed by the Interoperability Catalogue Data Model.

16-7.1 Geographic feature types

The relative prioritization for display purposes of geographic feature types is at the core of the Interoperability Catalogue Data Model. For the Interoperability Catalogue concept, feature types can be considered as the "domain" of the Data Model, just as feature concepts form the domain of an ordinary Product Specification.

Feature instances are not encoded in Interoperability Catalogues since an Interoperability Catalogue is a catalogue-based product that is functionally a collection of rules which adjust the display of information from feature datasets; an Interoperability Catalogue is not itself a feature-based data product.

References to feature types may appear as attribute values in Interoperability Catalogues. The reference will identify the product specification in which the feature type is defined. It may also identify the version of the Product Specification; if the version is not identified the reference is to the indicated feature type in all versions of the Product Specification.

A reference to a feature type must be interpreted as applying to all instances of the feature type in datasets conforming to the indicated Product Specification and version. (Additional conditions limiting applicability to subsets of feature instances may be encoded in other attributes.)

16-7.2 Meta feature types

The suppression, interleaving, and replacement operations in the Interoperability Catalogue do not affect meta features in individual Product Specifications. Display of meta features if requested by the mariner is as specified by individual Portrayal Catalogues.

16-7.3 Feature and Information associations

Feature and information associations are not directly used in an Interoperability Catalogue.

16-7.4 Information Types

Information types are not directly used in an Interoperability Catalogue.

16-8 Interoperability Levels

The interoperabilityLevel attribute (S100_IC_InteroperabilityCatalogue, S100_IC_DisplayPlane, S100_IC_PredefinedCombination, S-100_IC_HybridFeatureCreationRule) is used to indicate a level of interoperability. The value 0 means interoperability is off and all layers are portrayed according to their stacking order. When a value of 1 or higher is assigned to the interoperabilityLevel attribute it means the interoperability functionality is on. The exact meaning of any value of 1 or higher must be described in the interoperability catalogue specification where model is implemented.

Different product combinations may require different types of interoperability in order to work as intended in a system. For example, a product combination may require the ability to turn off a feature in product A to give prominence to a feature in product B. Another example may be that a product combination requires the ability to place product C in-between layers within product D. Within the same interoperability catalogue these different types of interoperability can be distinguished using different values in the **interoperabilityLevel** attribute of the classes that make up the functions.

Example: a specification has defined four levels of interoperability in addition to the so-called Level 0. The four levels start out with simple interleaving at Level One to virtually combing data at Level Four as seen in the example Figure 16-7 below.



Figure 16-7 - Example of how to separate interoperability functions into levels

16-9 Hybridization rules

Hybridization rules define how a set of feature instances are combined to create a hybrid feature type. In the simplest form of hybridization, and assuming that there are no collisions between the thematic attributes

of the input types, the hybrid feature would bind all the attributes of the input types to a single output feature type. More complex hybridization rules can handle collisions, for example by defining a preference order for colliding attributes, including all the values if the input types bind the same enumerated attribute, or adding uncertainty metadata if numeric attribute values are different. The hybridization rules require two feature instances as input and produce a single feature instance as output.

Note: The formal specification and rule language for hybridization will be described in a subsequent version of this document.

16-9.1 Simple hybridization rule

Simple rules treat thematic attributes uniformly, for example by binding the attributes of both primary and secondary input instances to the output instance, or preferring the attribute bindings of the primary instance to those of the secondary instance in case of a difference in the values of common attributes. Location/extent spatial attributes of all input instances must be spatially equal and are passed through unchanged.

16-9.2 Thematic hybridization rule

Thematic rules treat thematic attributes on an individual basis, for example, use specified attributes from the primary input instance and specified attributes from the secondary input instance. Combination operations on attribute values may be specified, for example, OutputFeature.depthValue = maximum(ProductA.FeatureX.depthValue, ProductB.FeatureY.depthValue. Location/extent spatial attributes of all input instances must be spatially equal and are passed through unchanged.

16-9.3 Complete hybridization rule

Complete rules allow selection of input sets using complex spatial queries as well as spatial equality and selector expressions on attribute values. The output can combine thematic attributes in any of the ways allowed by thematic hybridization rules. In addition, it may generate complex spatial objects from the input spatial primitives by applying selected spatial operations to the input instances. The allowed spatial operations will be identified in a future edition of this specification (tentatively, the spatial operations defined in IHO S-58, Section 2).

16-9.3 Hybrid Feature and Portrayal Catalogues

Hybrid Feature and Portrayal Catalogues are physically separate files from the main Interoperability Catalogue, but the main Catalogue links to them by encoding the names of the hybrid catalogue files which are used by the feature creation rules defined in it. The hybrid Feature and Portrayal Catalogues conform to the structures required by S-100 Parts 5 and 9 respectively. It is recommended that hybrid Feature and Portrayal Catalogues be bundled with the Interoperability Catalogue in an exchange set as support files, see Clause 16-10 below.

16-10 Production Process of Interoperability Catalogues

Interoperability Catalogues are created in an XML editor environment. It is generally expected that any offthe-shelf XML Editor can perform this task. The creation process of any new versions may benefit from starting from the previous version when available. Another alternative is to develop a dedicated Interoperability Catalogue Editor that can create and maintain interoperability catalogues with dedicated functions for creating the various components, such as display planes and predefined combinations.

Due to the interconnected nature of the Product Specifications that are under the interoperability schema, a form of overarching change management is a necessity. Any revision or New Edition required in a Product Specification should be announced well in advance, giving the whole stakeholder community ample time to review the impact before it goes into effect. Any revisions and New Editions to a supported Product Specification may require a new version of an Interoperability Catalogue and the party responsible for the maintenance of an Interoperability Catalogue needs to be informed and involved to assess any impacts.

This includes updates to dataset metadata, as metadata changes such as Product Specification references may impact the link between the dataset and the Interoperability Catalogue.

It is recommended that Interoperability Catalogues be distributed in an exchange set structure, see Part 4. Discovery metadata for Interoperability Catalogues must be provided in XML format conforming to S-100 dataset discovery as specified in S-100 Part 4a, with the extensions described in this clause.

16-10.1 Management of Feature Catalogue and Portrayal Catalogue updates

Changes to a supported Product Specification may have impacts on an Interoperability Catalogue. Revisions to the Feature Catalogue or Portrayal Catalogue are the most likely to require a revision of the Interoperability Catalogue in order to support the change. These types of changes will generally require a new version (n.n.0) of the Interoperability Catalogue to ensure support. It should be noted that revisions to a supported Feature Catalogue or Portrayal Catalogue may be ignored by previous versions of an Interoperability Catalogue and it is therefore necessary to consider this as part of the change management process, especially if the change is a matter of navigational safety. Major changes to Product Specifications, such as adding functionality or adding new Product Specifications to the supported list will result in a new edition (n.0.0) of an Interoperability Catalogue.

Versions of an Interoperability Catalogue within the same edition are considered a compatible group. When a New Edition is issued, this compatibility is broken; and efforts should be undertaken to update all impacted systems as soon as possible.



Figure 16-8 - Examples of how Feature Catalogue change may impact Interoperability Catalogue lifecycle

Several types of changes to supported Product Specifications may impact an Interoperability Catalogue in such a way that a new version is needed to maintain full support. These include:

- New feature added to a supported Product Specification that requires a new feature combination to be added to an Interoperability Catalogue;
- New attribute added to a feature in a supported Product Specification that requires a new attribute combination to be added to an Interoperability Catalogue;

- New Product Specification is added to the list of supported Product Specifications in an Interoperability Catalogue. This could also require new feature and attribute combinations to be added;
- Removal of feature or attribute from a supported Product Specification that are present in a feature or attribute combination within an Interoperability Catalogue;
- A correction to a supported Product Specification that triggers a version increment (n.n.0), which may break links from an Interoperability Catalogue to the supported Product Specification;
- Matters of navigational safety as they arise.

Adding new functions in either a supported Product Specification or an Interoperability Specification may require a new version of an Interoperability Catalogue.

16-10.2 Product Specification updates other than FC/PC

Updates to dataset metadata, such as Product Specification references, may impact the link between a dataset and an Interoperability Catalogue and therefore require a revision to the Interoperability Catalogue (n.n.0). Some changes to an interoperability-ready Product Specification may not require any changes to an Interoperability Specification or an Interoperability Catalogue. This includes amendments to the definitions of features, attributes or attribute values. It also includes minor changes to Product Specifications, such as clarifying language. Other changes may only require an update to the metadata of an Interoperability Specification and/or Interoperability Catalogue, such as in case of supported Product Specification version references. Such minor changes to an Interoperability Specification and/or Interoperability Catalogue and Interoperability Catalogue may be collected and be applied at a later time when a more substantial revision is required.

16-10.3 Unpredictable Product Specification updates

Unpredictable changes to an interoperability-ready Product Specification, or its Feature Catalogues and Portrayal Catalogues, should be avoided. Great care should be taken in coordinating changes among all stakeholders to avoid any unforeseen consequences. Product Specifications that are under the same interoperability umbrella should be considered interconnected. The responsible groups should therefore coordinate changes with other groups that issue interoperability-ready Product Specifications. Such coordination can be done, for example, by having an agreed fixed period between releases in which all pending changes are collected, implemented and issued in a coordinated fashion.

If a situation arises where a product is not compatible with an Interoperability Catalogue, this product will only be available as an overlay that may obscure any data below the incompatible product, or the overlay may be obscured by any data with higher priority.

16-10.4 Additional data products to be defined in catalogue

An Interoperability Catalogue should in theory be extensible to products not yet defined, provided these are within the same S-100 Edition. New products can be included in a Catalogue, but such scenarios will require a New Edition of an Interoperability Catalogue. The default processing for any Interoperability Catalogue is that any new product follows the default rule of being used as an overlay, generally stacked in a user defined order and with default viewing groups in effect.

16-10.5 Backward compatibility

Different versions of data products may be simultaneously active; the Interoperability Catalogue design allows for backward-compatible updates if and when the Interoperability Catalogue has to be updated,

within the same major Edition. Figure 16-9 below shows an example of how the Interoperability Catalogue may evolve with change over time.



Figure 16-9 - Example life cycle of Interoperability Catalogue

16-10.6 Interoperability Catalogue product

Each Interoperability Catalogue product should by itself be considered a whole unit. New versions – either clarification, correction, or New Edition – should be updated by replacement with a newer version. The update process should be specified in detail when defining an Interoperability Specification. The following clauses give some principles to consider for the specific update process.

16-10.6.1 Updating the Interoperability Catalogue

It is recommended that new versions of the Interoperability Catalogues be distributed using an exchange set structure, see Part 4.

Interoperability Catalogues may remain active after a new version has been issued. Systems receiving new versions within the same major edition should retain all versions. Store these in separate folders to avoid any issues, such as when the same support files have been reused between versions.

Due to issues with broken backwards compatibility, all previous versions of the Interoperability Catalogue should be cancelled when a New Edition is issued.

16-10.6.2 Cancelling a version of the Interoperability Catalogue

In order to cancel a version of the Interoperability Catalogue it is recommended that the methodology used for exchange set as defined in Part 4 metadata is utilized.

16-10.7 Support files

Support files to an Interoperability Catalogue are generally the hybrid Feature Catalogues and hybrid Portrayal Catalogues. Methods for managing these are described below.

16-10.7.1 Updating the Interoperability Catalogue support files

It is recommended that support files are updated using the methodology used for exchange set as defined in Part 4 metadata is utilized.

Support files should be stored in a separate folder within the exchange set.

16-10.7.1.1 New Edition of the support files

New Editions of the support files introduce significant changes. New Editions enable new concepts, such as the ability to support new functions, or the introduction of new constructs. New Editions are likely to have a significant impact on either existing users or future users of an Interoperability Catalogue Specification.

EXAMPLE: A new product is added to Interoperability Catalogue, and all support files should be updated to support the new product. This would require a New Edition of the support files.

16-10.7.1.2 Revisions to the support files

Revisions are defined as substantive semantic changes to the support files. Typically, revisions will change the support file to correct factual errors; introduce necessary changes that have become evident as a result of practical experience or changing circumstances. A revision must not be classified as a clarification. Revisions could have an impact on either existing users or future users of an Interoperability Catalogue Specification. All cumulative clarifications must be included with the release of approved revisions.

Changes in a revision are minor and ensure backward compatibility with the previous versions within the same Edition. Newer revisions, for example, introduce new feature or attribute combinations. Within the same Edition, a support file created for an Interoperability Catalogue of one version could always be processed with a later revision.

EXAMPLE: Adding a new hybrid feature will require a revision increment to the support file.

16-10.7.1.3 Clarification to the support files

Clarifications are non-substantive changes to the support file. Typically, clarifications: remove ambiguity; correct grammatical and spelling errors; amend or update cross references; insert improved graphics in spelling, punctuation and grammar. A clarification must not cause any substantive semantic change to an Interoperability Catalogue Specification.

Changes in a clarification are minor and ensure backward compatibility with the previous versions within the same Edition. Within the same Edition, a support file created for an Interoperability Catalogue of one version could always be processed with a later clarification (or revision) of an Interoperability Catalogue.

EXAMPLE: Correcting a spelling error in a definition will require a clarification increment to the support file.

16-11 Portrayal

This clause gives guidelines and instruction to portrayal considerations related to the use of the Interoperability Catalogue in a user system. An Interoperability Catalogue must apply to the specific Product Specifications listed in its metadata, specifically the **productCovered** attribute under **S100_IC_InteroperabilityCatalogue**.

There may be additional data products present in the user system that are external to the Interoperability Catalogue; in such cases the Interoperability Catalogue should continue to function in presence of product not defined in the Catalogue. Data products that are outside of the interoperability scope must be treated as overlays (see clause 16-10.4).

16-11.1 Display of significant features

Significant features in a display plane should have the highest drawing priority value within the **S100_IC_DisplayPlane**. Care should also be given to assigning significant features with high drawing priority values within the relevant **S100_IC_Feature** and **S100_IC_DrawingInstruction**. This ensures that less significant features in one data product are not displayed more prominently than more significant features in another product.

16-11.1.1 Switching to original

User systems should include means to easily switch on and off the interoperability functionality.

16-11.2 Impacts on viewing groups

The viewing group is a concept to control the content of the display. It works as an on/off switch for any drawing instruction assigned to the corresponding viewing group. The concept can be seen as a filter on the list of drawing instructions [S-100 Part 9, clause 9-11.1.3].

Viewing groups in interoperability catalogues take precedence over the applicable viewing groups for any feature instances from a supported Product Specification, and that are included in a **S100_IC_DrawingInstruction**, or **S100_IC_Feature** instance.

16-11.3 Impacts on Portrayal Catalogues

Viewing systems must manage the visibility and drawing order of data products, especially relative to radar/ARPA, AIS display or other sensor data. Moreover, systems must ensure significant features with over radar flag, in all products, are distinguishable in the presence of radar/ARPA, AIS or other sensor data. In navigation systems tracks and vessel position information are likely high priority while some AIS ASM (application specific messages) may carry lower-priority information, including data described by an S-100 based Product Specification, for example meteorological and hydrographic information.

16-11.4 Meta-features

It is recommended that viewing systems allow display of meta features for only one product at a time. This is in order to minimize risks of display clutter, user confusion, and the possibility of interpreting meta-features for one product as applying to a different product.

For example, data quality meta features for different on-screen products should not be displayed simultaneously, and that only the top most product data quality should be shown at any given time. This also applies in areas of the screen where the topmost product does not cover.

16-11.4.1 Data quality for individual products

This clause applies to the case where multiple products are on-screen and quality meta-features are enabled. Only one set of quality features should be displayed at any given time to avoid clutter and misreading the meaning of the quality metadata.

Interoperability Catalogues do not specify means of distinguishing data quality portrayals for individual products. Product Specifications must provide rules for display of data quality metadata (including data quality meta-feature information), which the user system will utilise to portray data quality.

Means of distinguishing data quality portrayals for individual products is left to the Product Specification authors (in particular, Portrayal Catalogue authors) and OEMs, and can be handled by distinguishing portrayal rules or symbology for different products' data quality meta-features, such as colour coding or special line symbol. There should also be a clear on-screen message saying what data quality features are displayed in order to give users a firm indication of the layer to which the currently displayed quality metadata applies.

16-11.4.2 Portrayal of data quality for combinations

Interoperability Catalogues do not include combining data quality portrayals. The recommendation in clause 16-10.4 about displaying only one set of meta-features is strengthened for data quality in particular. It is recommended that simultaneous portrayal of data quality from different products be avoided.

It is therefore recommended that user systems include functions to let the user select which product's data quality should be displayed.

16-11.5 Display of text

Text is typically the last item drawn. In general, rules for placement, display selection, and management of long text are defined in the individual product specification and associated portrayal catalogue. An Interoperability Catalogue would, in general, only govern when a feature that text is generated from is displayed.

Developers of user systems with interoperability capabilities should consider adding functions for enhanced automatic text placement to limit clutter.

16-11.6 Blended feature concepts or blended portrayals

This clause describes the interoperability solution for blended feature or blended portrayal. These can be produced by using transparency or creating a temporary blended feature or blended portrayal (rule and/or symbol) of specific combinations of features from different products. Such blended concepts will typically be created by using **S100_IC_PredefinedCombination** which links to a hybrid portrayal catalogue that includes the features to be combined and a suppression rule (using **S100_IC_SuppressedFeatureLayer**) for the features that are to be replaced.

An example where a blended concept could be used is where winds blowing from the west cause fairways to some west coast ports of Finland to get layered ice (wind pushes ice layers on top of each other until there is ice from the sea bottom up to the surface). When an ice-breaker makes a path through some ice remains between the sea bottom and the keel of the ice-breaker. Ice thickness in such a place could be up to 11 m while the ice-breaker draught is around 7-8 meters. In such cases a simultaneous display of both ice coverage and underlying depth area is required. Other depth area features such as spot soundings, rocks, wrecks, etc., are also still important.

16-11.7 Blended portrayal's effect on symbols and area patterns

When combining various layers, that may be of different compilation scale and coverage, it is likely that symbols and area patterns will end up at borders, or conflict with symbols and area patterns in other layers. It is important that symbols remain legible, and that user systems include appropriate methods to avoid displaying partial symbols, or "grafting" part of lower-layer symbols onto symbols in upper layers.

For area fills the symbols of a pattern fill must be closer together for a small or narrow area, to ensure enough symbols are seen, and farther apart for a large area, to avoid clutter. An area pattern may be substituted by a single centred symbol if sufficient space is not available for a pattern to be shown.

16-11.8 Hierarchy of data

Hierarchy between different Product Specifications is influenced by several factors such as intended use within the in-scope user system, for example usefulness a product in navigational operation in a navigation system. It may not be possible to prescribe a fixed hierarchy list as a universal standard, and the Interoperability Catalogue model therefore offers a flexible approach. Within the Interoperability Catalogue the hierarchy of data between different S-100 based Product Specifications is determined using predefined combinations. The **S100_IC_DisplayPlanes** referenced within a **S100_IC_PreDefinedCombination** give the order of feature layers. This approach also allows for different stacking orders of the same products

within the same Interoperability Catalogue should there be a user need for this. Care should be used (for example by using descriptive names and allowing users to see these when selecting) to ensure the risk of users unintentionally selecting the wrong stacking order.

16-11.9 Interacting gridded information

If two or more gridded data types are to interact, the hierarchy between them should be established using predefined combinations as with other data types. Particular care has to be taken depending on how the presentation of the data is to be done when deciding which gridded data type has the highest priority, considering items such as will one gridded data type obscure the other.

For example, gridded bathymetry may obscure gridded surface currents and therefore the gridded surface currents should be given the highest priority between the two when they are to be displayed simultaneously.

16-11.10 Pick reports

Pick reports may be defined in the individual Product Specification. The Interoperability Catalogue model permits reuse of these specifications as it does not specify pick report design for the individual supported Product Specification.

Complete data from all products visible on the screen should be available to the system user, irrespective of all these products being in the scope of the Interoperability Catalogue in use or not.

Features that have been visually suppressed should not be included in the pick report.

16-11.11 User control over loaded set

It is recommended that users have the option to load additional products in scope for the system, even when these are out of scope for the Interoperability Catalogue, or turn off one or more of the data products in a predefined combination. Portrayal must adjust to the loaded set as appropriate, for example if an additional product is loaded, it should be interleaved with layers from data products in the predefined combination according to the drawing order and drawing priority in its Portrayal Catalogue.

16-11.12 User control over interoperation level

If more than one interoperability level is supported by the Interoperability Catalogue (see 16-8), it is recommended that users have the option to select the interoperability levels they wish to use. Portrayal must adjust to the new interoperation level.

Appendix 16-A Interoperability Catalogue XML Schemas

(normative)

16-A-1 XML schema

16-A-1.1 Overview

Developers of S-100 Interoperability Catalogues must develop their data format consist with the Interoperability Catalogue XML Schemas referenced in this appendix. The schemas have a root or container element **S100_IC_InteroperabilityCatalogue**, whose structure is shown in Figure 16-A-1 below.

16-A-1.2 Unknown Attribute Values

When a mandatory attribute code or tag is present but the attribute value is missing, it means that the producer wishes to indicate that this attribute value is unknown. Missing mandatory attributes must be "nilled". Optional attributes must be omitted altogether if the value is unknown or missing. They must not be "nilled". This rule also applies to metadata.

16-A-2 Interoperability Catalogue Structure

An Interoperability Catalogue contains header information identifying the catalogue (a specialization from **CT_Catalogue** and extended with specific elements appropriate to S-100 interoperability, such as digital signatures and elements identifying the source of interoperability requirements). The header is followed by a list of products covered by the catalogue and containers for display planes, predefined combinations, hybridization rules and references to the hybrid portrayal and feature catalogues used by the results of hybridization rules. The general structure is depicted in Figure 16-A-1.



Figure 16-A-1 - Basic structure of an interoperability catalogue

The *displayPlanes* container is a collection of 1 or more **S100_IC_DisplayPlane** elements. The *predefinedProductCombinations* container is a collection of 0 or more **S100_IC_PredefinedCombination** containers. (The difference in the lower bound arises from the fact that it is possible for a catalogue to implement only interleaving but no higher level of interoperability.) The containers are depicted in Figure 16-A-2 and Figure 16-A-3 below.





Figure 16-A-3 - Container for predefined product combinations

The elements in each **S100_IC_DisplayPlane** and **S100_IC_PredefinedCombination** correspond to the model described in Clause 16-4 and are depicted in Figures 16-A-4 and 16-A-5 below. The contents of these elements are:

- S100_IC_DisplayPlane element:
 - the interoperability level;
 - the priority for the plane (displayPriority);
 - o identifying and descriptive elements (identifier, name, description);
 - o containers for **S100_IC_Feature** and **S100_IC_DrawingInstruction** elements.
- S100_PredefinedCombination element:
 - the interoperability level;
 - a list of the data products covered by this predefined combination (includedProduct elements);

- references to S100_IC_DisplayPlane elements;
- o optional containers for feature suppression rules and feature derivation rules;
- o identifying and descriptive elements (identifier, name, description, useConditions).



Figure 16-A-4 - Structure of individual display plane element



Figure 16-A-5 - Structure of individual predefined combination element

The structures of **S100_IC_Feature**, **S100_DrawingInstruction**, and **S100_IC_SuppressedFeatureLayer** elements correspond to the model and documentation in Clause 16-4 and are depicted in Figure 16-A-6, Figure 16-A-7 and Figure 16-A-8 below.



Figure 16-A-6 - Structure of S100_IC_Feature

Figure 16-A-7 - Structure of S100_IC_DrawingInstruction



Figure 16-A-8 - Structure of S100_IC_SuppressedFeatureLayer

The S100_IC_FeatureDerivation class is abstract and has two derived classes named S100_IC_SuppressedFeatureInstance and S100_IC_HybridFeature. The elements in each S100_IC_SuppressedFeatureInstance and S100_IC_HybridFeature correspond to the model described in Clause 16-4 and are depicted in Figures 16-A-9 and 16-A-10 below. The contents of these elements are:

- S100_IC_SuppressedFeatureInstance element:
 - primary product selectors;
 - secondary product selectors;
 - o output product indicators;
 - o identifying and descriptive elements (identifier, name, description);
 - reference to optional S100_IC_SimpleRule.

• S100_IC_HybridFeature element:

- primary product selectors;
- secondary product selectors;
- o output product indicators;
- o identifying and descriptive elements (identifier, name, description);
- o references to one of S100_IC_CompleteRule or S100_IC_ThematicRule.



Figure 110-A-9 - Structure of S100_IC_SuppressedFeatureInstance



Figure 16-A-10 - Structure of S100_IC_HybridFeature

The S100_IC_HybridFeatureCreationRule class is abstract and has three derived classes named S100_IC_CompleteRule, S100_IC_ThematicRule and S100_IC_SimpleRule. The elements in each S100_IC_SuppressedFeatureInstance and S100_IC_HybridFeature correspond to the model described in Clause 16-4 and are depicted in Figures 16-A-11, 16-A-12 and 16-A-13 below. The contents of these elements are:

- S100_IC_CompleteRule element:
 - o the interoperability level;
 - o identifier.
- S100_IC_ThematicRule element:
 - the interoperability level;
 - o identifier.
- S100_IC_SimpleRule element:
 - o the interoperability level;
 - o identifier.



Figure 16-A-11 - Structure of S100_IC_CompleteRule



Figure 16-A-12 - Structure of S100_IC_ThematicRule



Figure 16-A-13 - Structure of S100_IC_SimpleRule

An example of the XML conforming to the structures is shown in Figure 16-A-14. This figure shows the higher-level structure consisting of the catalogue header elements (*cat:name* through *requirementType*), followed by exemplary display plane and predefined product combinations. The lower-level containers *features*, *drawingInstructions*, and *suppressedFeatureLayers* are shown expanded in Figure 16-A-15 and Figure 16-A-16.

```
<$100IC:$100_IC_InteroperabilityCatalogue xsi:schemaLocation="http://www.iho.int/$100/IC ../../../.schemas
xmlns:lan="http://standards.iso.org/iso/19115/-3/lan/1.0" xmlns:gco="http://standards.iso.org/iso/19115/-5
xmlns:S100IC="http://www.iho.int/S100/IC">
  + <cat:name>
  + <cat:scope>
  + <cat:versionNumber>
  + <cat:versionDate>
  + <cat:language>
    <description>Example of interoperation catalogue structure</description>
    <comment>This is a hypothethical example showing only the structure of an interoperability catalogue file. 1
         actual features in the product(s) indicated.</comment>
    <requirementType>IHO</requirementType>
<productCovered codeListValue="S-101" codeList="resources/098AA00DICTIONARY.XML"/>
    version codeListValue="S-102" codeList="resources/098AA00DICTIONARY.XML"/>

    displayPlanes>

         <S100_IC_DisplayPlane>
             <identifier>1</identifier>
             <name>base</name>
             <order>1</order>
             <description>new SOE</description>
           + <features>
           + <drawingInstructions>
      </sloa_IC_DisplayPlane>
+ <S100_IC_DisplayPlane>
+ <S100_IC_DisplayPlane>
+ <S100_IC_DisplayPlane>
    </displayPlanes>
    <predefinedProductCombinations>
         <S100_IC_PredefinedCombination>
              <identifier>urn:mrn:iho:iop:0.1:pdc:1</identifier>
             <name>ENCandCurrents</name>
<description>combined S-101 and S-111 - suppressing all ENC current info</description>
             <useConditions>all conditions</useConditions>
              <interoperabilityLevel>2</interoperabilityLevel>
             <includedProduct codeListValue="S-101" codeList="resources/098AA00DICTIONARY.XML"/>
<includedProduct codeListValue="S-111" codeList="resources/098AA00DICTIONARY.XML"/>
            suppressedFeatureLayers>
         </S100_IC_PredefinedCombination>
     </predefinedProductCombinations>
```

```
</S100IC:S100_IC_InteroperabilityCatalogue>
```

Figure 16-A-14 - Example of Interoperability Catalogue showing the higher level structure



Figure 16-A-15 - S100_IC_Feature and S100_IC_DrawingInstruction - XML examples

```
<predefinedProductCombinations>
   <S100 IC PredefinedCombination>
      <identifier>urn:mrn:iho:iop:0.1:pdc:1</identifier>
      <name>ENCandCurrents</name>
      <description>combined S-101 and S-111 - suppressing all ENC current info</description>
      <useConditions>all conditions</useConditions>
      <interoperabilityLevel>2</interoperabilityLevel>
      <includedProduct codeListValue="S-101" codeList="resources/098AA00DICTIONARY.XML"/>
      <includedProduct codeListValue="S-111" codeList="resources/098AA00DICTIONARY.XML"/>
      <suppressedFeatureLayers>

    <S100_IC_SuppressedFeatureLayer>

             <identifier>urn:mrn:iho:iop:0.1:sfl:1</identifier>
             <featureCode>CurrentNonGravitational</featureCode>
             cycodeListValue="S-101" codeList="resources/098AA00DICTIONARY.XML"/>
             <featureRef>urn:mrn:iho:s111:1.0:CURENT</featureRef>
          </S100_IC_SuppressedFeatureLayer>
       </suppressedFeatureLayers>
   </S100_IC_PredefinedCombination>
```

Figure 16-A-16 - S100_IC_PredefinedCombination and S100_IC_SuppressedFeatureLayer - XML examples

Figure 16-A-17 shows an expanded example of a display plane showing one of the display planes for S-101 – this one contains navigation aids. The specification for one feature has been expanded as an example.

The drawing priority and viewing group given in a Portrayal Catalogue can be overridden for interoperation purposes by different values encoded in an Interoperability Catalogue, on a per-feature (type) basis.



Figure 16-A-17 - Example of display plane with S-101 features (informative)

Figure 16-A-18 depicts an example of the use of the substitution capabilities of an Interoperability Catalogue. The element substituteSymbolization for S-101 feature CurrentNonGravitational has its point and line symbols replaced by new symbols which are identified inline. The symbol and linestyle files are included in the Portrayal Catalogue identified by IHOICPCEXMP0001.



Figure 16-A-18 - Example of substitution in a drawing instruction (informative)

16-A-3 Location of schema files

The schema files are available from the locations listed in Table 16-A-1 below.

| Table | 16-A-1 - | Schema | files | locations |
|-------|----------|--------|-------|-----------|
|-------|----------|--------|-------|-----------|

| File | Description | Version | Location |
|-------------|--|---------|-------------------------------|
| S100_IC.xsd | XML schema for Interoperability Catalogue | 0.4 | https://github.com/IHO-S100WG |
| S100_IC.sch | Schematron file for validating Interoperability Catalogue | 0.4 | https://github.com/IHO-S100WG |

NOTE: The XML schema as distributed imports ISO metadata schemas from the ISO Internet location encoded in the S100_IC.xsd file. Implementers may wish to store the ISO metadata schemas locally and use the local installation instead.

Appendix 16-B

Guidance on how to Make Product Specifications Interoperable

(informative)

16-B-1 How to Make Product Specifications Interoperable

This appendix gives guidelines for how to identify concepts that should be factored into the development of an Interoperability Catalogue and suggestions for how to write rules to address associated interoperability issues. Moreover, these guidelines can be useful in the development of Product Specifications that will be included in an Interoperability Catalogue and its specification, in order to make these ready for interoperability in user systems.

Due to the interconnected nature of the Product Specifications that are under an interoperability schema, a form of overarching change and content management is a necessity. It is recommended that a management structure aimed at harmonization of Product Specifications that are under an interoperability schema is established.

For portrayal considerations, see Clause 16-11.

For interoperation requirements for Product Specifications, see Clause 16-C-3.

16-B-2 Duplicated features

Perhaps the most significant issue to deal with when considering interoperability is how to deconflict duplicated features between layers. The following paragraphs deal with major categories of duplicate features.

16-B-2.1 Duplicated features same model

Where there are equivalent models with same feature concept and attribute bindings, there still may be different attribute values due to issues with maintaining the same update sequence between different products. Both the production and end user sides can impact this issue. When considering this issue in the creation of an Interoperability Catalogue, priority should be given to the product that is most likely to be up to date with the latest information.

Developers of Product Specifications that are expected to be used in a system in interoperability mode should consider if the features within the specifications are likely to be more frequently updated than those

IC_SuppressedFeatureLayer elements only have feature code and product as attributes for suppression, this means that all instances of a listed feature class will be suppressed. This is important to remember when creating rules that promote alternative instances. IC_Feature and IC_DrawingInstruction can have attribute combinations and spatial primitives to select the alternative instances. There is therefore, a risk that unless sufficient attention to details is given, important instances may be omitted.

products that may serve as a base layer or base layer combination for the product being developed. These considerations should be factored in when describing the production of the product and envisioned future use of the product.

EXAMPLE: If **Restricted Area Navigational** in one product is suppressed, and **Restricted Area Navigational** with attribute **category of restricted area = 4 (nature reserve)** in another product is promoted in its place, there is a risk that only instances with that combination will be visible, and all others supressed.

16-B-2.2 Duplicated features, different models

Where the feature concept, attribute bindings, and values of selected attributes are mostly the same but there are minor differences in the different products, such as extra attribute bindings, developers of an Interoperability Catalogue should consider which version of the feature is the higher value for the end user, and give that version priority. There may be different answers depending on the operational situation that a predefined combination tries to support, and this must be considered as part of constructing an Interoperability Catalogue.

EXAMPLE: Interoperability Catalogue developers compare the specifications and data samples of **Pilot Boarding Place** features from a "Piloting Information" data product and the Pilot Boarding Place features in another product and decide that features from the "Piloting Information" datasets have more value in approach and harbour entry scenarios.

Developers of Product Specifications should consider how their data model is similar and/or different from other related Product Specifications and the justification for this; and make recommendations to the developers of an Interoperability Catalogue for how to best select between the versions of related features.

16-B-2.3 Duplicate feature domains

Where feature concepts are different, but the information content is equivalent, considerations should include the update cycle of the information and when creating an Interoperability Catalogue, priority should be given to the concept that is most likely to be updated most frequently. Other considerations should include any relations that the concepts has with other feature concepts, and any consequences of breaking these must be considered when choosing which concept to give the priority and which concept to suppress.

EXAMPLE: Developers investigate the update cycles of real-time current data products and discover that they are updated more frequently than a chart product with **Current – Non-Gravitational** and **Tidal stream – Flood/Ebb** features, and features from the real-time current datasets are therefore preferred over the chart product current features. Note that the question is decided not by comparing dates encoded in features, but on the basis of real-time data that is available on an ongoing basis vs. historical information gathered at a past date.

Where it is not practical to avoid concept overlaps, Product Specification developers should strive to maintain a data model that is as harmonized with related data models as possible. Due considerations should be taken before developing a concept that is different but functionally equal to similar concepts in other product specifications.

16-B-2.4 Geometry

The geometry of a feature is a significant attribute that must be considered when developing an Interoperability Catalogue, as it defines the 'where-part' of the feature object. Similar to other attributes, decisions may have to be made to address issues such as select one feature over another, for example where types in one product is affected by small scale, while another by large scale, or if merging the two is a better approach. The following paragraphs give more details about options for deconflicting geometry between products with the help of an Interoperability Catalogue.

16-B-2.4.1 Combined geometry

Where there is a feature in one dataset that effectively augments the geometry of a conceptually different feature in another dataset, developers of an Interoperability Catalogue need to specify a hybrid feature with

portrayal that can correctly portray the combined information. Moreover, it should be considered if it is appropriate for clarity in the user system to supress the features in the origin datasets.

EXAMPLE: A dredged area augmented with high definition bathymetry from survey of recent dredging operations giving more water and wider area than dredged area in a navigation product, combine to give a bigger (new boundary) dredged area than present in the navigation product.

Developers of Product Specifications that may result in hybrid features when interacting with certain other products, should cooperate with the developers of the related Product Specifications to correctly define the conditions for appropriate use of hybrid features, and communicate these specifications to the Interoperability Catalogue developers.

16-B-2.4.2 Spatial discrepancy, unrelated to scaled or cartographic smoothing

If the same feature instances exist in different data products, and are expected to have discrepant geometries, developers of an Interoperability Catalogue should establish the cause of the discrepancy. The cause will likely influence the solution to be implemented in the specific Interoperability Catalogue. For example, if one of the data products has more detailed information due to differences in the scopes of the Product Specifications, the developers of an Interoperability Catalogue should consider suppressing the feature class in the less-detailed product giving preference to the same feature class in the product with greater detail. On the other hand, if the discrepancy is found to occur irregularly, it may be more appropriate to suppress the less-detailed instances and prefer the more-detailed instances, independently of the data products to which they belong.

Example: A navigation product has Restricted Area features, while an environmental product has Marine Protected Area features which show greater details for a sub class of restricted areas.

Developers of Product Specifications should examine their Specification scope and consider if it is likely that resulting data products include information that will be superior or inferior to the same, or similar, information in other products. For example, if information is only for contextual purposes, it is likely that better information is available in another product and in an interoperability ready system these contextual features should be supressed in the presence of more accurate information. Such expectations should be communicated with the Interoperability Catalogue developers where the products are included.

16-B-2.4.3 Spatial discrepancies, related to scale or cartographic smoothing

Where there is the expectation of differences of geometry for same feature instance in different products it is important to establish the cause, as this will likely impact the solution implemented in an Interoperability Catalogue. If the cause is related to scale or cartographic smoothing in one product over another due to issues such as different scopes (intended use) of the Specifications, the Interoperability Catalogue developers should consider supressing the lesser detailed product and promoting the product with greater detail. If it is irregular occurrences of spatial discrepancies, it may be appropriate to supress the instances where there are less detail and promote the more detailed instances.

EXAMPLE: A navigation product has approximate surface current instances using climatic data that is omitted at smaller scales; A surface current gridded data product has greater details and is daily updated. The surface current information in the navigation product may be suppressed to give preference to the information from the surface current gridded data product

Developers of Product Specifications should examine the Specification scope and consider if it is likely that resulting data products include information that will be superior or inferior than the same information in other products. For example, if information is only for contextual purposes or is reduced in detail due to scale, it may be that better information is available in another product and in an interoperability ready system these lesser detailed features should be supressed in the presence of more accurate information. Such expectations should be communicated with the Interoperability Catalogue developers where the products are included.

16-B-2.5 Display of text

For details about display of text, including placement, display selection and management of long text, see Clause 16-11.5. The Interoperability Catalogue does not address these issues in general terms.

Developers of Product Specifications should be aware that the instructions they place within the Product Specification generally carry through even when the product is used in a user system in interoperability mode. Moreover, text placement issues in interoperability mode are expected to occur at the border between two products which may result in text being partially obscured due to priority issues. Developers of user systems with interoperability capabilities will likely have experience with solving such issues and advice should be sought with them in how to mitigate such issues. Additionally, Part 16-11.5 recommends OEMs to provide functionality that seek to address most of the issues that cause text to be partially obscured.

16-B-2.6 Skin-of-the-earth replacement

Some systems implement a skin-of-the-earth concept that require the system to always provide continuous coverage of select data elements to make up a surface of the earth. These data elements are usually drawn first in portrayal processing. In some circumstances, it may be necessary to replace some of these select data elements with more appropriate data. In the interoperability model, skin-of-the-earth replacement is a specialization of combined geometry, see Clause 16-B-2.4.1 for details.

16-B-2.6.1 Skin-of-the earth feature adjusting

When combining feature instances into new skin-of-the-earth feature instances, considerations should be given to the attributes of the resulting skin-of-the-earth feature, as a combined feature may have altered geographical representation, attribute combinations or attribute values that may impact its inclusion in the skin-of-the-earth.

EXAMPLE: Shoaling in a channel in a navigation product may be indicated by a high definition bathymetry product, and a shallower channel hybrid feature replaces the navigational product feature instance, which also has an amended shape. Adjacent depth areas grow due to the shoaling.

16-B-2.7 Blended feature concepts

Blended feature concepts or blended portrayals can be produced by using transparency between related features; or creating a temporary blended feature; or blended portrayal (rule and/or symbol) of specific combinations of features from different products. See 16-11.6 for portrayal considerations and example of use case. Blended concepts will typically be created by using **S100_IC_PredefinedCombination** which link to a hybrid Portrayal Catalogue that includes the features to be combined and a suppression rule, for example by using **S-100_IC_SuppressedFeatureLayer**, for the features that are to be replaced.

Developers of Product Specifications that are likely to be used in blended feature concepts by ECDIS in interoperability mode should communicate their intentions with developers of related Specifications so that awareness is created about the inter-dependencies of these types of relationships. Such communication is especially important when revisions to these Specifications are considered. Doing so will help manage risks to breaking the relationships as the related Product Specifications transition through their life cycle.

16-B-2.8 Hierarchy of data

In this context, hierarchy of data means the stacking of data products (layers) within a predefined combination. Predefined combinations are generally created with a particular type of operational view in mind, and therefore the hierarchy of data may vary between predefined combinations.

For example, in ECDIS, the ENC will be the base layer; that is, the lowest layer in a predefined combination.

16-B-2.8.1 Predefined combinations

Predefined combinations are used to define the hierarchy of data between different S-100 based specifications. An instance of **IC_PredefinedCombination** is associated to **IC_DisplayPlane** instances to give the hierarchy of the data products that are intended to be used. The attribute **displayPriority** within the **IC_DisplayPlane** gives the order in which the layers are drawn. See 16-4.4.2.6 for more details on predefined combinations.

16-B-2.9 New datasets

New datasets that are added to a system with interoperability mode will be managed in accordance to any loaded interoperability catalogue that includes the Product Specification that relevant data product is conformant with. Data producers should therefore be encouraged to perform sufficient tests to ensure new datasets perform as envisioned in an interoperability environment.

16-B-2.9.1 New datasets - coverages

New datasets may alter the available coverages of particular data that is used for interoperability views; and any new dataset should therefore be sufficiently tested to ensure performance is as envisioned.

16-B-2.9.2 New datasets - maximum and minimum display scales

New datasets may alter the available data in particular scales and/or scale bands, for example, by adding or removing data coverage. Considerations should therefore be given to harmonization of maximum and minimum display scales when a new dataset is produced.

16-B-2.9.3 New datasets - feature geometry

New datasets may alter the available feature geometry of available data that is used for interoperability views. Changes include extending or reducing size of areas, changing geometry type from area to point, point to area, area to line or line to area. Therefore, any new dataset should be sufficiently tested to ensure performance is as envisioned.

16-B-2.9.4 New datasets - types and attributes

New datasets may change type and attributes of instances in the user system, for example a platform may be removed and an obstruction remain. These changes may impact the situational view created by the Interoperability Catalogue as changes to feature classes and attribute combinations may mean objects are no longer covered by conditions specific to a predefined combination, or new objects are new covered. Therefore, any new dataset should be sufficiently tested to ensure performance is as envisioned.

16-B-2.10 Dataset loading and unloading due to scale

Developers of Product Specifications and producers of data should make every effort to harmonize effects of maximum and minimum display scales at loading/unloading time between related product to control over-scale indicators and datasets, in order to avoid situations where one overlay is in scale but not another.

16-B-2.12 New data products

When a new product is added to an existing Interoperability Catalogue, a new version will be required, see 16-10.4. During the development of the new version, the Interoperability Catalogue developers will need to review existing predefined combinations for impact in addition to developing the new predefined combinations to manage the situational views that the new product is intended for. Close coordination between the authors of the Product Specification and the Interoperability Catalogue producer is highly recommended to ensure all relevant changes to the Interoperability Catalogue are accounted for.

16-B-2.13 Metadata

This edition of the Specification does not provide for comparing information that is not encoded as attributes of feature (or information type) instances. This means that metadata cannot be used in interleaving, filters, or rules unless it is encoded in feature attributes (for example "horizontal position uncertainty" attributes) or meta-features (for example **Quality of Bathymetric Data**). If Product Specification authors envisage a need to use metadata in interoperability, the Product Specification Application Schema should be designed so as to make the relevant metadata available as feature attributes or meta-features.

Note that using information from meta-features in interoperability operations may involve spatial operations.

16-B-3 Portrayal distinguishability - colour set-asides

Special consideration must be made when creating portrayal rules related to colour choices for a Product Specification. Depending on the context certain colours have specific meaning. For example, for marine navigation, IHO S-4 gives indications for magenta line meaning something non-physical, while black colour implies a physical item. See IHO S-4, clauses B-141 to B-145 for additional details.

Example: on ENC the light sectors marking intricate inshore channels in, such as in Scandinavian waters are shown in red, green, and yellow.

16-B-3.1 Black (S-4 B-141)

Black is normally used for all physical (solid) features, including depth information.

16-B-3.2 Magenta (S-4 B-142)

The general principles for the use of magenta are that it should be reserved for:

- Drawing attention to symbols for features which have a significance extending beyond their immediate location.
- Distinguishing information superimposed on the physical features and not implying any permanent physical obstruction (see S-4 clause B-145 for the use of green for environmental information).

16-B-3.3 Buff (yellow) or grey (S-4 B-143)

A colour, usually buff (yellow) or grey, must be used as a land tint in paper charts. ENC portrayed with S-52 in an ECDIS uses a yellow/brown colour (LANDA).

16-B-3.4 Blue (S-4 B-144)

The colour blue has been used as a tint to emphasize shallow water. Two (or more) densities of blue tint may be used to show different depth bands of shallow water, the darkest tint showing the shallowest water.

16-B-3.5 Green (S-4 B-145)

The colour green may be used as a tint for inter-tidal areas. Green may also be used, instead of magenta, for environmental information and limits; see S-4 clause B-437.2b.

16-B-3.6 Red

It should be noted that any symbology using red may be an issue for navigation equipment operated in night mode, and therefore the use of red should be avoided as much as possible.

16-B-3.7 Day/night/dusk modes

It may be required that for every Product Specification that is intended for a navigation system to have colours specified for day, dusk, and night modes. The system may be required to utilize these colours depending on the mode the viewing system is set to.

16-B-4 Rendering order steps

Product Specifications that are in scope of an Interoperability Catalogue should have at least ten rendering order steps between display groups. This allows more flexibility when configuring interleaving with other products in an Interoperability Catalogue.

Appendix 16-C User Interaction Constraints and Expectations

(informative)

16-C-1 Introduction

S-100 based products are developed for a variety of uses but generally define only the product itself and its use, not how it will be used in in combination with other S-100-based data products or sensor information. It is anticipated that most S-100-based systems, like ECDIS and other ship and shore-based systems, will use several different products simultaneously, each providing one or more information layers. Other data layers such as real-time sensor information from sources like radar and AIS are also expected to be present. The smooth interoperation and harmonized user-friendly graphical presentations of these various products is necessary for safety and efficiency. This annex gives some guidance to principles that should be considered when developing smooth interoperation and harmonized graphical presentation of S-100 data products

16-C-2 Structured interoperation

IMO has issued recommendations and guidelines on how to present navigation related information. Of particular significance are the documents noted below.

MSC.191(79), Recommendation on Performance Standards for the Presentation of Navigation-Related Information on Shipborne Navigational Displays, specify the presentation of navigational information on the bridge of a ship, including the consistent use of navigational terms, abbreviations, colours and symbols, as well as other presentation characteristics. It also addresses the presentation of navigation information related to specific navigational tasks by recognizing the use of user selected presentations in addition to presentations required by the individual performance standards adopted by the Organization.

MSC.1/Circ.1512, Guideline on Software Quality Assurance and Human-Centred Design for E-Navigation, gives guidance in achieving trustworthy software and usability in the development of complex systems requires a disciplined and structured approach. The guideline encourages particular focus on Software Quality Assurance (SQA) and Human-Centred Design (HCD) that includes Usability Testing (UT). Systems so designed, developed and managed throughout their life cycle deliver improved user performance, being stable and resilient, and, most importantly, support users in low and high workload environments, such as during challenging navigation and environmental conditions when users are most vulnerable to making mistakes and when error management and recovery is essential. Other important benefits include limiting the amount of operator familiarization training that is needed and the time and resources required for system maintenance and support.

MSC.1/Circ.1609, Guidelines for the Standardization of User Interface Design for Navigation Equipment, apply to Integrated Navigation Systems (INS), Electronic Chart Display and Information Systems (ECDIS) and radar equipment. The guidelines may also be useful to other electronic navigation equipment, and navigation sensors to improve standardization and usability. The aim of the Guidelines is to promote standardization of user interfaces to help meet user needs.

IMO SN.1/Circ.243/Rev.2, Guidelines for the Presentation of Navigational-Related Symbols, Terms and Abbreviations, stems from a compelling user need for greater standardization to enhance usability across navigation equipment and systems. Significant variation between systems and equipment produced by different manufacturers has led to inconsistency in the way essential information is presented, understood and used to perform key navigation safety functions. Improved standardization of navigation systems will provide users with more timely access to essential information and functions that support safe navigation.

16-C-3 Interoperation requirements for Product Specifications

Responsible parties for Product Specifications that are included in an Interoperability Catalogue should consider the impact on this Interoperability Catalogue and associated Product Specifications throughout the lifecycle of the Product Specification. The general principles of Software Quality Assurance (SQA) as found in section 4 of the Annex to MSC.1/Circ.1512 should be applied.



Figure 16-C-1- Generic life cycle (from MSC.1/Circ.1512)

Figure 16-C-1 shows a typical generic life cycle with the stages recommended as a minimum for the management of the development of Product Specifications that are used with the Interoperability Catalogue:

- Analysis of operational system feedback;
- Stage 1: Concept development;
- Stage 2: Planning and analysis;
- Stage 3: Design;
- Stage 4: Integration and testing;
- Stage 5: Operation; and
- Disposal.

16-C-4 Customization - user level

S-100 compatible systems that support the IHO Interoperability Catalogue may include functionality that allow end users (including ship owners, operators, and shipboard officers) to add new predefined combinations according with their needs. These added combinations must not interfere or degrade the official IHO Interoperability Catalogue functions.

16-C-5 Support Human-Centred Design

As noted in MSC.1/Circ.1512, HCD helps to ensure that human factors-related knowledge and techniques in system design and development processes are addressed, thus ensuring that user needs and safety are met. Implementers of this Specification should perform Usability Testing (UT) and follow the principles

stated in MSC.1/Circ.1512 when designing the user interface for interoperability in ECDIS, including the following HCD activities that are carried out to inform development throughout the life cycle:

- Pre-activity: Conduct Early Human Element Analysis (EHEA);
- Activity 1: Understand and specify the context of use;
- Activity 2: Identify the user requirements;
- Activity 3: Produce and/or develop design solutions to meet user requirements;
- Activity 4: Evaluate the design against usability criteria; and
- Activity 5: Maintain operational usability.

Note that fundamental to HCD is the collection of user feedback through Usability Testing.



Figure 16-C-2 - Overview of HCD for e-navigation systems (from MSC.1/Circ.1512)

The details of recommended activities at each stage are found in section 6 of the Annex to MSC.1/Circ.1512.

16-C-5.1 Human-Centred Design and user interface

MSC.1/Circ.1609, Guidelines for the Standardization of User Interface Design for Navigation Equipment notes in section 15 principles utilized in the formation of the guidelines. These and the resulting appendixes may be of benefit for any system implementing support for an Interoperability Catalogue.

16-C-5.2 Human-Centred Design and Text

MSC.191(79) as amended, notes in section 5.2.3 that use of text should be presented using simple unambiguous language that is easy to understand. Navigation terms and abbreviations should be presented using the nomenclature defined in IMO SN.1/Circ.243/Rev.2 and MSC.1/Circ.1609 and encouraged their

use for all shipborne navigational systems and equipment. Shore based equipment that interact with shipborne system may also benefit from guidance in these guidelines.

16-C-6 User control over loaded set

Users of systems with interoperability capabilities may have functionality to load additional products, or turn off one or more of the data products in a predefined combination.

Data from such additional products which are not mentioned in an Interoperability Catalogue should be treated by the system according to the priorities and viewing groups encoded in the product's own Portrayal Catalogue (for example interleaved with layers from products controlled by the Interoperability Catalogue according to their relative rendering orders and drawing priorities).

When a user turns off a data product, the portrayal should treat it as if the relevant datasets are not available on the system at all. For example, interoperability rules that are made inapplicable due to one of the data products in their conditions being turned off are unavailable.

16-C-5 User control over interoperation level

The system should allow the user to change the interoperation level (see 16-8) and/or pick a predefined combination by means of simple operations. Any options offered to the user must be valid in context; for example, if the interoperability catalogue supports more then one interoperability level, the user interface should offer the user a choice of predefined combinations at an interoperability level. The listed combinations should be only those defined at that level in the Interoperability Catalogue.

The system should minimise demands for user interaction when changing interoperability level or predefined combinations, subject to constraints imposed by the platform and interface. Some implications of this guidance are:

 When the interoperability level alone is changed and the Interoperability Catalogue contains a predefined combination of the new level that lists the currently displayed product set, the system should apply the rules of the new level to the product set immediately. Alternative predefined combinations for the level may be offered in an unobtrusive way.

EXAMPLE: Product A and Product B data are both on-screen when Level 1 (no suppression) is changed to Level 2 (suppression supported), and the catalogue includes a "Level 2 Product B + Product A" predefined combination. Interleaving of Product A and Product B features (Level 1) immediately changes to suppression of Product A navigation aids by Product B navigation aid features (Level 2).

Optionally, an indicate may be provided to inform the user of the active predefined combinations defined at the new level in the Interoperability Catalogue.

2) When the predefined combination alone is changed and the Interoperability Catalogue contains the new predefined combination at the current level, the system should apply the rules of the current level to the new predefined combination.

EXAMPLE: The system is in Level 2 (suppression supported) and the Product B + Product A predefined combination is changed to Product B + Product A + Product C. The system suppresses Product A Restricted Area features of type "nature reserve" in favour of Marine Protected Areas from Product C.

3) When the Interoperability Catalogue does not contain a predefined combination at a newly selected level, the user interface should provide an indication of this to the user (though not necessarily by disabling the choice or blocking the transition). Strategies for dealing with this situation are left to interface designers. For example, systems may offer to use the closest fit in an Interoperability Catalogue with any residual on-screen products as ordinary overlays.

16-C-6 Priority overrides for user-specified settings

Where user action amends a setting, which then conflicts with a system setting, the user setting should override the system setting. The system may give indication of this override.

EXAMPLE: Feature rendering order set by a user should override settings in an Interoperability Catalogue or Portrayal Catalogue.

Appendix 16-D Data Encoding Guide (Normative)

16-D-1 Introduction

This appendix contains encoding guidance on syntax, content, and catalogue structure for Interoperability Catalogue developers. Guidance on how to make Product Specifications interoperable and what principles to apply when developing an Interoperability Catalogue is provided in 16-B. For definitions of catalogue elements, their attributes, and associations, refer to 16-4.

16-D-2 General encoding notes

16-D-2.1 Identifiers and references

Several catalogue elements have an *identifier* attribute. The value of this attribute must conform to the syntax for a Uniform Resource Identifier (URI). This means it may be a URL, an integer, alphanumeric character string without whitespace, or a URN. Any additional restrictions are mentioned in the encoding notes for the appropriate element, which make up the rest of this appendix.

Some catalogue classes in the catalogue data model (16-4) have associations that act as references to the element at the other end of the association. This is encoded in the data format as an XML child element of the referrer, whose XML tag is the same as the role name in the UML model. The value of such a reference must be equal to the value of the *identifier* attribute of the referenced element.

EXAMPLE:

Thetag<featureRef>urn:mrn:iho:chart:CURENT</featureRef>inanS100_IC_SuppressedFeatureLayer element is a reference to the S100_IC_Feature element with tag

<identifier>urn:mrn:iho:chart:CURENT</identifier>.

16-D-2.2 Feature codes

Some catalogue elements have a *featureCode* attribute. The value of attribute *featureCode* must be the camel case code of the feature as encoded in the Feature Catalogue for the product named in the *product* co-attribute.

16-D-2.3 Element S100_IC_InteroperabilityCatalogue

Any product mentioned in any attribute of type *dataProduct* of a catalogue element (for example **S100_IC_Feature.***product*, **S100_IC_PredefinedCombination.***includedProduct*, etc.) must also be mentioned in a *productCovered* attribute of this element.

The attribute *productCovered* will be used by the system in deciding whether to apply interoperability rules or fall back on ordinary overlay portrayal and therefore all products taken into account when developing the catalogue must be listed.

The name, description, and requirementDescription attributes should be populated with text values of appropriate size that are meaningful to people developing, configuring, or using Interoperability Catalogues, including end users. These attributes, especially requirementDescription, will potentially be displayed to end users when they select an Interoperability Catalogue for a particular task supported by a system, and should be populated with this use in mind.

The attribute *interoperabilityLevel* may be used to indicate highest level of interoperability functionality encoded within the catalogue.

16-D-2.4 Element S100_IC_DisplayPlane

A **S100_IC_DisplayPlane** element must contain at least one instance of **S100_IC_DrawingInstruction** or **S100_IC_Feature**. It may contain multiple instances of either or both. The choice depends on whether symbols or other components of drawing instructions are being substituted.

The *name* and *description* attributes should be populated with text values that indicate the purpose and feature content of the display plane to people developing, configuring, or using Interoperability Catalogues, including end users.

16-D-2.5 Element S100_IC_DrawingInstruction

Drawing instruction elements in the feature catalogue override the drawing instructions generated directly from the data product's Portrayal Catalogue. Details of this overriding are described in 16-4.2.2.4.

The attributes *product, featureCode, geometryType*, and *attributeCombination* values together make up a <u>filter condition</u> determining the subset of instances of a feature type to which the drawing priority and viewing group encoded in **S100_IC_DrawingInstruction** apply. They are to be applied in conjunction ("AND") – that is, the **S100_IC_DrawingInstruction's** viewing group and drawing priority apply only when the conditions expressed by all these attributes are satisfied. (Attributes *geometryType* and *attributeCombination* being optional are ignored if not encoded.)

If an instance of **S100_IC_DisplayPlane** contains both **S100_IC_Feature** and **S100_IC_DrawingInstruction** elements with the same "filter condition", their drawing priority and viewing group must also be the same.

Features (drawing instructions) not satisfying a filter condition in an instance of **S100_IC_DrawingInstruction** are treated according to any other interoperability rules which may apply to them, or if none, they treated according to the data product's portrayal catalogue.

Distinction: S100_IC_Feature.

16-D-2.6 Element S100_IC_Feature

S100_IC_Feature elements in the Interoperability Catalogue override the drawing priority and viewing group in the data product's portrayal catalogue. Details of this overriding are described in 16-4.4.2.3.

S100_IC_Feature elements have the same four filter condition attributes as **S100_IC_DrawingInstruction** and the same rules and constraints described in 16-D-2.4 apply.

Distinction: S100_IC_DrawingInstruction.

16-D-2.7 Element S100_IC_DisplayPlane

As noted in 16-D-2.5 and 16-D-2.6 an instance of **S100_IC_DisplayPlane** must contain at least one of **S100_IC_Feature** or **S100_IC_DrawingInstruction**.

16-D-2.8 Element S100_IC_PredefinedCombination

A **S100_IC_PredefinedCombination** includes at least two different data products. The attribute *includedProduct* must be populated with all data products referenced directly or indirectly in this predefined combination, including:

- The *product* attribute of a **S100_SuppressedFeatureLayer** contained in this element;
- The *product* attribute of a S100_Feature or S100_IC_DrawingInstruction referenced by an S100_IC_SuppressedFeatureLayer element contained in this element.
- When a **S100_SuppressedFeatureLayer** is included, a reference to the replacement **S100_IC_DrawingInstruction** should also be present.

The name, description, and useConditions attributes should be populated with text values of appropriate size that are meaningful to people developing, configuring, or using Interoperability Catalogues, including end users. These attributes, especially *useConditions*, will potentially be displayed to end users when they select an Interoperability Catalogue for a particular task supported by a system, and should be populated with this use in mind.

16-D-2.9 Element S100_IC_SuppressedFeatureLayer

The *featureCode* and *product* attributes identify a feature type which will be suppressed in its entirety; that is, all instances of the feature from that product will be hidden. They will be replaced by instances of the feature type and product indicated by the referenced **S100_IC_Feature** (or **S100_IC_DrawingInstruction**). Both elements may include conditions pertaining to attribute values and geometry type, as described in clause 16.5. The implications should be carefully considered when referencing instances of **S100_IC_Feature** or **S100_IC_DrawingInstruction**, for example:

- Consider whether all feature instances of the indicated feature type from the replacing product will be displayed, or only a subset selected by attribute values.
- Consider what happens to the excluded instances and if these should be displayed or not. (Note that the model allows replacement of one feature type by multiple feature subsets, for example a Chart layer Ice Area may be replaced by multiple ice features from a Sea Ice layer).
- Consider whether the spatial attributes change and implications if they do. For example, does the replacing product include features of both point and surface while the replaced product includes only point features.

16-D-2.10 Element S100_IC_SuppressedFeatureInstance

The **S100_IC_SuppressedFeatureInstance** element is used by **S100_IC_PredefinedCombination** for selecting instances from one product to be replaced by instances from another product.

The attributes *primaryProduct*, *primaryFeatureCode* and *primarySelector* specify the primary feature and product, which is to be suppressed.

The attributes *secondaryProduct*, *secondaryFeatureCode* and *secondarySelector* specify the secondary feature and product, which should replace the suppressed instance(s).

The attributes *outputProduct* and *outputFeatureCode* specify the replacement feature and product, and should be identical to the secondary product. The output feature does not need to have any hybrid characteristics, see 16-4.2.2.7.

The reference to a **S100_IC_Feature** gives the resulting feature's display specification in the display planes section of the interoperability catalogue.

The optional reference to a S100_IC_SimpleRule gives the rules describing how a set of feature instances are combined to create a hybrid feature type.

Where two different set of features instances are to be suppressed by one common feature instance, two instances of S100_IC_SuppressedFeatureInstance is needed.

Distinction: S100_IC_HybridFeature.

16-D-2.11 Element S100_IC_HybridFeature

Complex operations, including conversion of input feature instances into new features (hybridization) are described by associated **S100_IC_HybridFeature** elements. The replacement and hybridization rules are described in Clause 16-4.

The attributes *primaryProduct*, *primaryFeatureCode* and *primarySelector* specify the primary inputs that will be suppressed from rendering.

The attributes *secondaryProduct*, *secondaryFeatureCode* and *secondarySelector* specify the secondary inputs that will be suppressed from rendering.

The attributes *outputProduct* and *outputFeatureCode* specify The attributes *outputProduct* and *outputFeatureCode* specify the replacement feature and product, which may be defined in a hybrid FC and hybrid PC.

The reference to a **S100_IC_Feature** gives the resulting feature's display specification in the display planes section of the interoperability catalogue.

The reference to either a **S100_IC_ThematicRule** or a **S100_IC_CompleteRule** gives the rules describing how a set of feature instances are combined to create a hybrid feature type.

Distinction: S100_IC_SuppressedFeatureInstance.

16-D-2.12 Element S100_IC_CompleteRule

This class is used for rules describing how a set of feature instances is combined to create a hybrid feature type. Complete rules may operate on both thematic and spatial attributes.

16-D-2.13 Element S100_IC_ThematicRule

This class is used for rules describing how a set of feature instances is combined to create a hybrid feature type. Thematic rules may operate on only thematic attributes. The input features are required to have spatially equal geometry within a tolerance set by the system.

16-D-2.14 Element S100_IC_SimpleRule

This class is used for rules describing how a set of feature instances is combined to create a hybrid feature type. Simple rules may operate only on thematic attributes. Simple rules treat thematic attributes uniformly during hybridization and therefore do not mention specific attributes. Location/extent spatial attributes of all input features must be spatially equal.

16-D-2.15 Element S100_IC_HybridFC

Class for holding a reference to a Feature Catalogue defining any "hybrid feature types" that are created by combining feature types from two or more products for the purposes of an interoperable display. The Feature Catalogue must conform to S-100 Part 5.

16-D-2.16 Element S100_IC_HybridPC

Class for holding a reference to a Portrayal Catalogue defining portrayal rules for the "hybrid feature types" defined in a Hybrid Features Catalogue. The Portrayal Catalogue must conform to S-100 Part 9.

Appendix 16-E

Implementation Guidance

(informative)

16-E-1 Implementation Guidance

This appendix provides guidance on implementation issues that should be considered when developing an Interoperability Catalogue for an end-user system. Individual interoperability specifications may elaborate and modify this guidance as appropriate.

16-E-2 Reduce demand on user attention - display adjustment

It is recommended that implementers of end user systems supporting interoperability catalogues include decluttering techniques, such as minimizing overlaps of both symbols and text and minimization of the number of colours simultaneously on the display.

16-E-3 Reduce demands on user attention - avoid text overload

It is recommended that implementers of end user systems provide means for text notes to be shown in a manner that limits the obscuring other for example by including for separate text display from graphic display.

It is recommended that means are provided for limiting the amount of text shown simultaneously both ingraphic, over-graphic, and in a separate auxiliary display.

Where possible it is recommended that in-line text is kept shorter than text from a support file. This may be done through limiting the allowed text in some attributes in product specifications. For example, allow as many as 300 characters in any attribute intended for in-line text.

Interoperability Catalogue developers should review text handing of in scope product specifications, including what individual Data Classification and Encoding Guides say and what Portrayal Catalogues do with text attributes. This review should be done in collaboration with Product Specifications developers, since they can be expected to know which attributes can be expected to contain long text and which contain short text.

16-E-4 Support for novice users

It is recommended that end user implementations are permitted to have "novice" modes or user interface controls, which provide shortcuts for inexperienced users.

16-E-5 Reduce demands on user attention - planning and monitoring modes in navigational systems

Navigational systems with planning mode should be permitted to provide more powerful information search or processing functionality at the expense of more user attention.

Navigational systems with route monitoring mode should provide means for showing the information required for route monitoring while allowing users to also focus on other tasks.

16-E-6 Interoperability and data coverage

The interoperability rules and interleaving operations described in an Interoperability Catalogue should only apply in areas where the products referenced in the rule or interleaving operation have data coverage at the current display scale on the navigation system.

If data coverage for some of the products in the selected predefined combination is absent in an area, the rules and interleaving operations referring to products which do have data coverage in the area in question should continue to apply in that area. Rules and interleaving operations referring to products which do not have data coverage in the area should not apply in the area in question.

Implementations in end user systems should be capable of indicating parts of the display screen where (a) interoperability is partially applicable because some of the data products in a predefined combination do not have data coverage while others do have coverage; (b) interoperability is not applicable at all because the data products in the selected predefined combination do not have coverage.

Note: Depiction and symbols for such distinguished parts of the screen may be a matter for Performance Standards but an off-graphic message on the system, or an adaptation of appropriate warning symbology may be suitable.

16-E-7 Other significant information

The inclusion in Interoperability Catalogues of data products whose interoperability has not been discussed with the relevant Product Specification development team is recommended <u>against</u>.

There should be a dialogue between interoperability teams and Product Specification teams, so that new changes to Product Specifications are ensured to be covered by Interoperability Catalogues.

Feature Catalogue and Portrayal Catalogue development teams should be considered stakeholders for hybrid catalogues.

16-E-8 Phased implementation

Implementation of interoperability may be done in phases and this can be done by utilizing the interoperability level attributes (see 16-8) to segment different functionalities of an interoperability catalogue.